WO02077855

Publication Title:

SYSTEMS AND METHODS FOR CONTENT DELIVERY OVER A WIRELESS COMMUNICATION MEDIUM TO A PORTABLE COMPUTING DEVICE

Abstract:

Abstract of WO02077855

A system which analyzes, organizes and stores electronic documents in a document object model using a common markup language. The document object model is stored on a server (10,14,50,52) and the content is delivered to the wireless device (12) in a universal content system format (162). By utilizing the document object model and the universal content stream with a corresponding reader on the wireless device, the electronic document is reduced to only the essential content and the necessary format information for transmission. Data supplied from the esp@cenet database - Worldwide

---
Courtesy of http://v3.espacenet.com

This Patent PDF Generated by Patent Fetcher(TM), a service of Stroke of Color, Inc.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
3 October 2002 (03.10.2002)

PCT

(10) International Publication Number
WO 02/077855 A1

(51) International Patent Classification[7]: G06F 17/21

(21) International Application Number: PCT/US02/09268

(22) International Filing Date: 25 March 2002 (25.03.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/278,137 23 March 2001 (23.03.2001) US

(71) Applicant: ARIZAN CORPORATION [US/US]; 430 10th Street, N.W., Suite S-105, Atlanta, GE 30318 (US).

(72) Inventor: SYLTHE, Olav, A.; 4867 Ashford Dunwoody Road #6208, Atlanta, GE 30338 (US).

(74) Agent: WINBURN, John, T.; Morris Manning & Martin, LLP, 3343 Peachtree Road, NE, Atlanta, GA 30326-1044 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

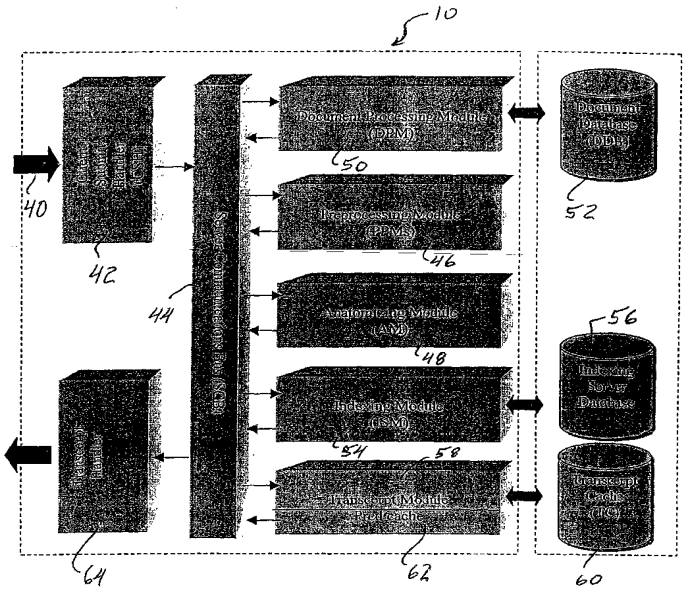
(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declaration under Rule 4.17:
— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii)) for all designations*

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SYSTEMS AND METHODS FOR CONTENT DELIVERY OVER A WIRELESS COMMUNICATION MEDIUM TO A PORTABLE COMPUTING DEVICE



(57) Abstract: A system which analyzes, organizes and stores electronic documents in a document object model using a common markup language. The document object model is stored on a server (10,14,50,52) and the content is delivered to the wireless device (12) in a universal content system format (162). By utilizing the document object model and the universal content stream with a corresponding reader on the wireless device, the electronic document is reduced to only the essential content and the necessary format information for transmission.

WO 02/077855 A1

SYSTEMS AND METHODS FOR CONTENT DELIVERY OVER A WIRELESS COMMUNICATION MEDIUM TO A PORTABLE COMPUTING DEVICE

CROSS REFERENCE TO RELATED APPLICATION

This application claims priority under 35 U.S.C. 119 to the benefit of the filing date of Sylthe U.S. patent application serial no. 60/278,137, which was filed on March 23, 2001, entitled "SYSTEMS AND METHODS FOR CONTENT DELIVERY OVER A WIRELESS COMMUNICATION MEDIUM TO A PORTABLE COMPUTING DEVICE," which is incorporated herein by reference.

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates generally to wireless content delivery of documents and more specifically relates to organizing document content to be requested by reconstructed and read on a wireless device.

Description of the Prior Art

With the rapidly expanding use of computers, electronic documents are stored and communicated between computers with relative ease. Full desktop systems and laptop systems readily transfer these documents over various communication media, such as the Internet.

It is also desirable, however, to utilize these documents with the wireless communications medium and utilizing mobile wireless devices. This presents a whole series of different problems, since the bandwidth available can be on the order of 4 to 48 kbps, which is not expected to increase past 56 kbps in the near term.

The wireless devices typically attempt to maximize battery life, while at the same time minimizing weight and the device physical size. Typically, the devices also include

slow processors and a minimal amount of memory. At the same time, the devices have different, generally limited, operating systems, screen resolution and limited graphic rendering capabilities. Large documents with a wide variety of formats are therefore very unsuitable or impossible for use with these devices in their standard format.

It would thus be desirable to provide a system, which allows the wireless devices to have access to the electronic documents without regard to their size or formatting of the documents.

SUMMARY OF THE INVENTION

The present invention provides a system, which analyzes, organizes and stores electronic documents in a document object model using a common markup language. The document object model is stored on a server and the content is delivered to the wireless devices in a universal content stream format. By utilizing the document object model and the universal content stream with a corresponding reader on the wireless device, the electronic document is reduced to only the essential content and the necessary format information for wireless transmission.

To accomplish the above and related advantages, this invention may be embodied in the form illustrated in the accompanying drawings, attention being called to the fact, however, that the drawings are illustrative only, and that changes may be made in the specific construction illustrated without departing from the scope of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

Various other functions, features and attendant advantages of the present invention will become fully appreciated as the same is described in conjunction with the accompanying drawings, in which like reference characters designate the same or similar parts throughout the several views, and wherein:

DETAILED DESCRIPTION OF THE INVENTION

Figure 1 is an illustration of a data processing system or server where the invention is employed to communicate with wireless devices of clients over the Internet.

Figure 2 is an illustration of a block diagram of the data processing system and wireless devices represented by **Figure 1**.

Figure 3 is a block diagram illustration of the invention residing on the server in **Figure 1** detailing individual process modules of the present invention.

Figure 4 illustrates the data packaging of a document submitted to the present invention.

Figure 5 is a flowchart illustrating the process flow of a submit request being processed by the data processing system on the server.

Figure 6 is a flowchart detailing the process flow of an incoming request being processed by the Server Communication Bus (SCB) module of the server data processing system.

Figure 7 is a flowchart detailing the process flow of a pre-processing module for invoking external processes for Optical Character Recognition (OCR), Image Enhancement and Raster to Vector conversion (R2V).

Figure 8 is a flowchart detailing the process flow of a data processing system Anatomization module used to construct the documents anatomy in establishing the Document Object Model (DOM).

Figure 9 is a flowchart detailing the process flow of an Indexing Module for indexing content of a submitted document.

Figure 10 is a flowchart detailing the process flow of a Transcerption Module of the invention for delivering the relevant content excerpts to the wireless/wired clients.

Figure 11 is a block diagram illustrating the invention residing on a wireless device or client detailing the individual data processing modules.

Figure 12 is a flowchart detailing the process flow of the invention residing on a wireless device receiving transcerpted data in the form of the universal content stream (UCS).

Figure 13 is a diagram illustrating the architecture of the processing engine of the present invention with a loaded structured or unstructured document or data stream.

Figure 14 is a diagram illustrating the relationship of a parent and child object in the invention of a constructed Document Object Model (DOM) for a loaded document.

Figure 15 is an illustration of a sample unstructured document with which the invention might be used.

Figure 16 is an illustration of a sample structured document with which the invention might be used.

Figure 17 is an illustration of the constructed Document Object Model (DOM) utilized by the invention for a loaded structured document.

Figure 18 is a diagram illustrating a Document Object Model of the present invention for textual content.

Figure 19A is a diagram illustrating the inventions document object model in constructing a paragraph of textual content.

Figure 19B is a continuation of illustrating the paragraph command in **Figure 19A**.

Figure 20 is a diagram illustrating the Document Object Model for a loaded document containing table data as typically found in a spreadsheet document type.

Figures 21A – 21F are illustrations detailing the command content of a transcerpt response from the invention to a wireless/wired device in the form of a Universal Content Stream data structure for a structured and unstructured document respectively.

Figure 22 is one embodiment of a layout for the communication format of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

A data processing system or server **10** communicating with one or a plurality of wireless devices or clients **12** where the present invention can be utilized is illustrated in **Figure 1**. The data processing system **10** typically includes conventional components, such as a central processing unit, internal mass storage, and random access memory, which are utilized by the present invention. Communication to and from the data processing system **10** will typically come through a secure software or hardware processing system/unit commonly known as a Firewall/Proxy Server **12** in a conventional manner to ensure authorized and valid access to the network in which the invention is deployed. The invention executing on the data processing system or server **10** will typically receive inbound and transmit outbound data via the Internet **16** to a portion of the invention residing on the wireless device or client **12**. A current embodiment of the present invention can utilize as the data processing system **10** DELL, COMPAQ, SUN or other recognized manufacturers of server hardware running a Windows operating system release NT 4/2000/XP or higher or on a Sun Solaris operating system release 8 or higher.

Figure 2 illustrates a more detailed overview of the data processing system or server **10** and the wireless device or client **12**. The server **10** includes a central processing unit (CPU) **20**, a random access memory (RAM) **22** and an internal mass storage device **24** all interconnected by a data channel commonly known as a bus **26.** The data processing system **10** communicates with the wireless device client **12** over the Internet **16.** The wireless device client **12** processing system includes a central processing unit (CPU) **28** designed for small footprint devices such as a Motorola Dragon Ball CPU, or the Intel StrongARM CPU coupled with a read only memory (RAM) **30** which stores a small footprint wireless device

operating system such as Palm OS, or PocketPC OS with an internal flash ram memory storage **32** communicating with the Internet **16** utilizing a wireless radio/modem **34** all coupled together using a databus **36**.

The data processing system **10** preferably is configured to execute and perform the functionality of the current embodiment of the present invention by having an operating system able to utilize data instructions written to execute using the functionality provided by the operating system to utilize typical programming languages such as C++ and Java.

Figure 3 illustrates a block diagram of the process flow for the invention's server side processing modules of the present invention. A module by definition is an individual data process optional or obligatory that combined with other modules forms the invention's data processing system **10**. A structured or unstructured electronic document or data stream **40** is submitted from the wireless or wired client side **12** and enters the server **10** from the network as binary data. A Client Submit Handler **42** receives the client request and verifies that this is a valid request and passes the received data with appropriate routing commands based on the client request over to a Server Communication Bus **44**, which reads the routing commands and sends the received data to the required processing modules in the required order as indicated by the routing commands. If the first routing command is a request for a Preprocessing Module (PPM) **46** the received data is passed along for document or data optimization. The Preprocessing Module's **46** main purpose is to improve document quality for utilization by the invention if the document type is a scanned document containing imagery depicting textual content, containing binary imagery representing a picture, or a scanned document depicting a drawing using line or circular drawing elements.

For a scanned document containing imagery depicting textual content a technique known as Optical Character Recognition (OCR) is employed with the purpose of extracting the textual characters for processing by the invention. Any OCR style software package such

as OmniPage from ScanSoft Corporation can be applied by the invention for this purpose. For imagery representing a picture the invention can apply commonly known image algorithms for improving image quality such as sharpen, blur, Gaussian blur, motion blur, intensity, contrast, gamma correct, invert, hue and saturation, histogram equalize, emboss, mosaic, median, average etc. For a scanned document depicting a drawing using line or circular drawing elements raster to vector (R2V) software is employed for the purpose of extracting drawing element and positioning information from the scanned drawing to reconstruct the submitted image as vector drawing data on the inventions wireless/wired client. Any R2V style software package such as VP HybridCAD from SoftTelec can be utilized by the invention for this purpose.

The Server Communication Bus **44** will then route the data **40** on to an Anatomization Module (AM) **48**, which is responsible for establishing the anatomy of the document and to populate the Document Object Model (DOM) as described in **Figure 17** and will as a dependant process utilize the SCB **44** to route the data to a Document Processing Module (DPM) **50** where an Engine as described in **Figure 13** will load the structured or unstructured document using a distiller module. If the SCB's **44** routing commands include a command to persist (save) the document data parsed by the DPM **50** the document data will be saved in a Document Database (DDB) **52**. The Document Database **52** is a connector to typical software packages commonly referred to as databases such as Microsoft SQL Server from Microsoft Corporation. Upon successful processing completion of the document by the DPM **50** the SCB **44** will issue a command back to the Anatomization Module **48** and the Document Object Model will be updated with the content and anatomy of the document. If an indexing request is part of the command stack of the SCB **44**, an Indexing Module (ISM) **54** will be invoked and the textual content available from the Anatomization Module **48** of the document will be indexed using commonly known software products such as Microsoft

Indexing Server and the results along with the associated references to the Document Object Module will be stored in an Indexing Server Database **56**.

Upon completion by the ISM **54,** the SCB **44** command stack will invoke a Transcerpt Module (TM) **58** (transcerpt = transmit excerpt) and the initial excerpt of content will be prepared for transmission by invoking the Anatomization Module **48** and requesting an initial Table of Content or Summary of the document based on the content available in the Document Object Module. Upon return of the transcerpt by the AM **48,** a copy of the initial transcerpt will be placed in a Transcerpt Cache (TC) **60** for repeat access purposes. If the SCB **44** command stack contains a Predictive Caching Module **62** request, the AM **48** will be repeatedly invoked for the lifetime of the client session with the purpose of predicting future client requests for related content data from the document. The Predictive Caching Module **62** will, for the lifetime of the client session, then populate the Transcerpt Cache **60** with the transcerpt results from the Transcerpt Module **58.** The initial transcerpt is then upon completion from the TM **58** returned via the SCB **44** to a Transcerpt Handler **64** where the correct client device **12** for the return data is identified and the transcerpt is sent to the network **16** and back to the wireless/wired client **12.**

If the Client Submit Handler **42** receives a client request for more content from a previously submitted document it passes the request over to the Server Communication Bus **44,** which in turn hands it off to the Transcerpt Module **58,** which will then look into the Transcerpt Cache **60** to see if a transcerpt generated by the Predictive Caching Module **62** is available for transmittal. If not, the Anatomization Module **48** will be invoked and a new transcerpt will be prepared following the procedure as described for documental submittal for the System **10.** Upon completion the transcerpt will be placed in the Transcerpt Cache **60** and the Trancerpt Handler **64** again will be invoked and deliver the transcerpt to the client wireless/wired device **12** as described for the initial document submittal.

Figure 4 represents the packaging of a document or content request **70** submitted for publishing by the present invention. An XML based metadata description **72** is the encapsulation of the packaging and will contain the appropriate server command such as publish or query, wireless device profile of the requesting device **12** providing information such as device type, device screen size, free storage space, estimated bandwidth transfer speeds etc. and information on the document **70** such as document name, document date, document type or any other supplemental information. An Original Document or unique document ID **74** for the content request **70** makes up the other part of a document submittal request. If an original document is part of the submitted request the document will be published or republished if required. If a unique document ID is sent as part of the request it indicates that the document is already published and the server will look in the Transcerpt Cache **60** or retrieve the document from the Document Database **52** based on the unique Document ID.

Figure 5 illustrates a flowchart process **80** for a wireless client **12** submit request to the data processing system server **10**. An incoming request **82** with data has arrived. The first step of the process **80** is to establish if this is a new publish request **84**. If it has a unique document ID **74**, it is not a new publish request (NO) and the document already resides on the server data processing system **10,** then a Transcerpt processing step **96** will take place directly. If an original document is being submitted, the metadata **72** is processed to see if there is a client request to perform Pre Processing **86** and the PreProcess module **46** will be invoked in a step **88.** If not, the Anatomization process **90** will be performed. Upon process completion from the Anatomization process module **48** the metadata **72** is processed to see if there is a client request to perform indexing **92** on the anatomized content. If this is true, then the Index Processing Module **54** is invoked and the content will be indexed in a step **94.** Upon completion of step **92** and/or step **94,** the Transcerpt module **58** will be invoked in a

step **96** with the purpose of delivering the initial content result back to the requesting wireless client **12**.

Figure 6 is a flowchart **100** illustrative the data flow of the server side data processing system **10** of the Server Communications Bus **44** (SCB). An incoming request and/or data is received **102** by the SCB and the SCB looks to determine if this is a valid request **104** for the SCB **44**. If the request is valid (YES), the command and associated data is retrieved **106** by the SCB. Associated data will be an original document or original document unique ID. Pending on the SCB command the data will be routed **108** to the appropriate processing module where any single or combination of SCB commands will be performed by the server side data processing system **10** including any of the system modules; the Document Processing Module **50**, the Anatomization Module **48**, the Preprocessing Module **46**, the Indexing Module, and the Transcerpter Module **58**. As the SCB **44** is designed to process both asynchronous or synchronous request requests for data processing by the individual data processing system modules **46, 48, 50, 54, 58** the requests may or may not be completed when the SCB **44** will check if there are additional commands and data to be processed in a step **110**. If there are more commands or data (YES) the process will continue again with the step **106** until the command stack is emptied. If no more data or commands are pending in the step **110,** the SCB **44** will return the appropriate response and resulting data from the server side data processing modules in a step **112** to the requesting wireless/wired client **12**. If no valid command is detected by the SCB in the step **104** (NO) the appropriate failure code will be assigned by the step **112** and the resulting return or response will be sent to the requesting wireless or wired client **12**.

Figure 7 illustrates a flow chart **120** describing the server side data processing system for the Pre Processing Module **76**. Inbound Data **122** is received on the SCB **44**. A validation step **124** is performed to see if this is a valid Pre-process command. If the

command is not recognized as a valid pre-process command (NO) the result error code is set to E_FAIL in a step **126** and the error code is returned as outbound data on the SCB in a step **128**. If the step **124** determined that the command is a valid preprocess command (YES), a step **130** checks to see if the document type submitted with the preprocess command is a valid supported document. If the answer is negative (NO) the step **126** again sets the result error code to E_FAIL and the error code is returned as outbound data on the SCB in the step **128.** If it is determined that this is a valid document type (YES) in the step **130**, then the Preprocess module **46** is invoked in a step **132**. Based on the document type determined as part of the successful step **130**, an appropriate preprocessing sub module **134, 136** or **138** is invoked. Step **134** performs an Optical Character Recognition (commonly referred to as OCR) to convert a scanned document into textual content. The invention is designed to use commercially available software for OCR purposes from software vendors such as OmniPage from ScanSoft Corporation.

Step **136** is invoked by the step **132** if the valid document type determined in the step **130** is an image or picture data in saved form such as BMP (Bitmap) or JPEG (Joint Photographic Experts Group) upon which commonly known mathematical formulas can be employed to improve the visual aspects of the image picture data such as sharpen, blur, Gaussian blur, motion blur, intensity, contrast, gamma correct, invert, hue and saturation, histogram equalize, emboss, mosaic, median, average etc. A step **138** is invoked by the step **132** if the valid document type determined in the step **132** is a scanned drawing depicting a drawing using line or circular drawing elements. A step **138** will then employ commercially available raster to vector (R2V) software for the purpose of extracting drawing element and positioning information from the scanned drawing to reconstruct or recreate the submitted image as vector drawing data on the wireless/wired client **12**. Any R2V style software package such as VP HybridCAD from SoftTelec can be utilized by the present invention for

this purpose. Upon completion of any sub process **134, 136, 138** by step **132** then a step **140** will check if the process is determined to be successful. If the step **140** is a success (YES), the preprocessed data will be returned back to the SCB in the step **128.** If the step **140** determines that the result from the step **132** is unsuccessful (NO), the result error code is set to E_FAIL in the step **126** and the error code is returned as outbound data on the SCB in the step **128.**

Figure 8 is a flow chart **150** illustrating the process flow of the server side data processing system **10** for the Anatomization process **48.** Inbound data from the SCB is received in a step **152,** which is passed to a step **154** where it is determined if the document data is valid or not. If the document data is invalid (NO) a step **156** sets the result error code to E_FAIL and the pointer to the DOM is set to equal zero or NIL, and returns the error code as outbound data on the SCB in a step **158.** If it is determined that the document data is valid (YES) in the step **154,** the document data received will be parsed in a step **160** and the Document Object model (DOM) then is built up in a step **162.** After the DOM has been constructed by the step **162,** then a step **164** will check if there is a command request to persist or store the constructed DOM. If the step **164** is true (YES) the DOM data will be stored in the Document Database **52** in a step **166** before continuing on to a step **168.** If the step **164** is false (NO), then the step **168** will check to see if there is a command to apply a decorator. A decorator is a command to traverse the Document Object Model established by the step **162** and extract content or navigational information such as a Table of Content (TOC), Bookmarks, Hyperlink, Find Text etc. If the step **168** has no decorator command to apply (NO), a step **170** will set the result code to S_OK and set the DOM pointer before returning the outbound data in the step **158.** If, the step **168** determines that decorators are to be applied (YES), a step **172** is invoked and the decorators will traverse the DOM established

in the step **162**. Upon completion of the step **172** the result code is set to S_OK and the DOM pointer is set in a step **172** before returning the outbound data in the step **158**.

Figure 9 illustrates a flow chart **180** of the Indexing Process **54** for the server side data processing system **10**. Inbound data from the SCB **44** arrives in a step **182**. A step **184** checks if there is a request to index the data. If the step **184** is determined to be false (NO), then a step **186** will set the result code to E_FAIL and the result will be returned as the outbound data to the SCB in a step **188**. If the step **184** is true (YES), then the system will start to traverse the Document Object Model (DOM) in a step **190**. In a step **192** the content of a single DOM element is retrieved and then the system **10** will start to build the Index in a step **194**. In the step **194** the system will typically employ commercially available software indexing packages such as Microsoft Indexing Server. After the content of the retrieved DOM element has been indexed in the step **194** then a step **196** checks to see if the DOM traversal is complete. If the answer is NO, then the steps **192** and **194** will again be processed until the DOM traversal is complete (YES) in the step **196.** The index is then stored in a step **198** in the Index Database **56** and the result code is set to S_OK in a step **200** before the result is returned as the outbound data to the SCB in the step **188.** The Index DB **56**, utilized in the step **198** can be any commercial database product such as MS SQL Server or Oracle.

Figure 10 is a process flowchart **210** of the Transcerption module **58** for the server side data processing system **10**. Inbound data arrives from the SCB in a step **212** and the data is checked to see if this is a transcerpt request or not in a step **214**. If it is not a transcerpt request (NO), then a step **216** will set the Result code to E_FAIL and pass the result back to the SCB in a step **218** as outbound data. If the step **214** determines that the data is a transcerpt request (YES), then a step **220** is invoked to obtain the ID of the Document Object Model. The DOM ID is used to identify the correct document either from the transcerpt cache **60** or from the Document Database **52**. A step **222** checks to see if the data is available

in the cache **60**. If the answer is YES, then a step **224** will get the data from the transcerpt cache **60** and a step **226** will set the result to S_OK and return the outbound data in the step **218** onto the SCB. If there is no data in the cache (NO), during the step **222** then a step **228** will see if there is a DOM loaded in the memory and if it matches the DOM ID.

Because the system **10** employs caching it will typically retain the DOM and DOM ID for a specified number of documents, such as the last 32 processed documents and if the DOM ID matches any of the DOM content in the cache it will be put back in memory unless it's already there, meaning it was the last document processed. If the test in the step **228** determines that there is no DOM in memory or the DOM ID does not match (NO) (meaning the DOM is not in the cache) then a step **230** is invoked, which will retrieve the DOM from the Document Database **52** before proceeding with a step **232**. The system **10** allows the DOM to be saved to a database or file for immediate retrieval without having to reprocess the original document used to create the DOM initially. If the step **228** determines that there is a DOM in memory and the Document ID matches (YES), then the step **232** is invoked to get the transcerpt. A step **234** then looks to verify if the SCB command received in the step **212** as inbound data from the SCB has the Predictive Caching Flag set. If the flag is set (YES), then a step **236** will start the asynchronous Predictive Caching process **62** before proceeding with a step **238**. If the step **234** determines that the flag is not set (NO), then the step **238** is invoked and the transcerpt data retrieved in the step **232** will be committed to the cache **60**. A step **240** will then test repeatedly to see if the commit is complete, returning back to the step **238** if the test results in a negative (NO). When the step **240** returns true for commit complete (YES) then step **224** is invoked and the transcerpt data is retrieved from the cache. The step **226** will set the result to S_OK and return the outbound data in the step **218** onto the SCB.

Figure 11 is a detailed diagram **250** of the present invention wireless client processing system residing on a wireless device or Personal Digital Assistant (PDA) **12**. A reader **252** is a software application including a plurality of individual software function elements and libraries **254** through **266** that makes function calls to a Wireless PDA/Device operating system **270**. The operating system **270** includes a plurality of individual software functions forming the operating system **270**. Commonly known operating systems (OS) for wireless PDA/devices are Palm OS by Palm Corporation, Windows CE by Microsoft Corporation, J2ME by Sun Corporation, EPOC by Symbian and BREW by Qualcomm. The inventions client side software functionality is designed to read the defined Universal Content Stream, also commonly referred to as UCS. The reader **252** has the Graphic Function Library **266** that offers a set of graphical instructions used to render textual, vector and image-based data on the device **12**. To accomplish this the library **266** uses available graphics operations **254** from the device OS **270**. The Core Function Library **264** offers a set of functions derived from the device OS **270** for the memory access and handling **272**, the file input/output and/or database access **274**, and the wireless data access **276** for retrieving data from the Internet **16**. The reader **252** then has the application command layer **262** that invokes the core library **264** and the graphic function library **266** to be able to offer a graphical user interface GUI as defined in the view modes **254, 256, 258** and **260**.

Based on the data loaded from the UCS the Image View **254** will display image or picture based data, the Sheet View **256** will display spreadsheet data in a spreadsheet style grid, the Document View **258** will display textual content in a rich and intuitive way and the File View **260** will display a user interface for loading and saving the UCS file on the device **12**. The layered design in the reader **252** is designed so that only the core function library **264** and the graphics function library **266** will be different on each wireless/pda device **12** on which the client side processing system is deployed on. This allows the invention to be

quickly moved or ported to different operating systems such as Palm OS, Windows CE, J2ME, EPOC and BREW where the two primary programming languages are Java and C++ as commonly know by those skilled in the art of programming. A USC binary file is delivered to the device **12** from the server side data processing system **10** over the Internet **16** and the wireless client reader **252** retrieves this data by using the functions for wireless data access (load) **276** in the core function library **264**, which in turn calls the library for remote HTTP/TCP/IP access **276** of the wireless PDA/device OS **270**.

Figure 12 is a detailed flowchart **280** of the client side processing application residing on the wireless PDA/devices **12**. In a step **282** the reader **252** is invoked either by being launched by another application on the device **12** such as the Inbox application or by the user selecting the application icon for the reader and invoking it. In a step **284** the passed Universal Content Stream (UCS) if any, will be verified to be local or remote. If local then the UCS is already stored on the device **12** and a step **286** will read the local UCS stream from a file or database **288**. If the UCS is determined to be remote in the step **287** (not residing on the device **12**) then a step **290** is invoked to read the remote UCS stream. A step **292** will then obtain the remote UCS residing on the server side data processing system **10** using the Internet **16**. Upon completion of either step **286** or **290** the execution will continue with a step **294** where the retrieved UCS data is parsed. A step **296** will then load the data in device memory and then a step **298** will test to see if the parsing of the remote or local UCS data is complete. If the step **298** is not complete for local data (NO) then the step **294** will be invoked again until the step **298** is true (YES) for local UCS files. If the step **298** for remote UCS parsing is not complete (NO) then a step **300** will create a remote link list for any UCS data described to still reside on the server **10** for any future transcerpts to the client. Then a step **302** updates the UCS memory with the remote links and continues with the step **294** until the remote parsing step **298** is complete (YES).

After the UCS data has been parsed and is loaded into memory, then a step **304** determines the view type or combination of view types for the UCS data. Depending on the view type either Image View **254**, Document View **258**, or Sheet View mode **256** is invoked and then the UCS data is displayed or rendered in a step **306**. After displaying the UCS data in the step **306** the client data processing application waits for user input in a step **308**. If the user navigates within the loaded document using menu options, or track/scroll wheel, stylus or keypad then the step **308** will be repeatedly invoked until the user terminates the reader **252** by choosing the exit option and a step **310** will exit the mobile reader and return to the invoking application prior to the step **282**. If the user selects or invokes a more request in the step **308** for additional UCS data, then the step **290** will be invoked again with the same following process flow as previously described until the user chooses to terminate the application in the step **308** and then the step **310** will exit the mobile reader and return to the invoking application prior to the step.

Figure 13 is a diagram **320** representing the document engine structure with a hypothetical document loaded for the document-processing module **50**. The composition of the document engine is a collection of individual components that are categorized as the following modules, an Engine Module **322**, a set of Distiller Modules **324** and a loaded document **326** and Decorator Modules (not illustrated). The distiller module **324** functionality is mainly to handle different document types. This set of modules is responsible in particular for parsing a known document format, constructing the document structure and, if desired, persisting the document structure back to a known document format. The set of distiller modules **324** is open in the sense that new distiller modules can be added or removed at runtime to accommodate the need for a flexible environment. So the set of supported document types can be expanded or reduced as required. The engine module **322** functionality is mainly to host the running distiller modules **324** and, if desired, the loaded

documents **326**, offering a uniform environment for these. It is also used to provide common behavior for clients (e.g. loading a document). A plurality of the document structure modules are the building pieces of a loaded document. They include: a document node **328** (representing the loaded document and containing all other nodes), contents nodes **330** (representing the contents with a certain detail degree of a document), a pair of container nodes **332, 334** (representing containers for other nodes), a text component **336** (representing textual content), an image component **338** (representing image content), a vector component **340** (representing vector content), and a table component **342** (representing table data such as spreadsheets). The document structure **326** is a tree-like structure (with the possibility of sharing of nodes, so the more appropriate description would be graph-like). This fact allows the user to practically construct documents in any fashion desired without being constrained by a rigid document structure.

This is achieved by the fact that each node implements collection like methods (for children and/or parents). Behavior like rendering can be attached to the node itself or be offered as an external component. This is facilitated by the existence of a set of standard iterators of the document structure. The choice between internal and external behavior can be taken by taking into consideration and the requirements of the intended operation (speed, size, deployment etc.) The decorator modules functionality (not shown in **Figure 13**) is mainly to perform customized selected operations on the document structure. Examples include the extraction of the table of contents, list of bookmarks or hyperlinks, find operations etc. It is here where the transcerption technology is employed. They are separate modules, with their own framework, that operate completely independent of the other engine components. In fact the engine can operate without the decorator modules being installed. They just add more functionality. The decorator modules framework is composed of a decorator manager (which is by itself a component) and a set of installed modules that

perform certain operations. The individual modules have to register in a predefined way in order to integrate into the framework and be recognized by the decorator manager. A decorator is a software component as commonly referred to by those skilled in the art of software development designed for the sole purpose of navigating the Document Object Model (DOM) to retrieve specific content and/or navigational information. In other words it can be stated that a decorator component constitutes a command. There are no limitations on how many decorators can be used to navigate the DOM. Typical decorators (commands) are: retrieve bookmarks in a document, retrieve hyperlinks in a document, retrieve a Table of Content for a document, construct a summary for a document if no TOC is available, retrieve a specific page, find textual content within a document, retrieve specific font styles or information, retrieve an image or illustration within a document, etc.

Figure 14 illustrates the relationship **350** of a parent **352** and a child node **354** within the document engine **320**. The actual link between the parent **352** and the child **354** is a little more complex than a pointer between the two. Between them there is a so-called specifier **356** object whose role is to hold attributes of the link. By attributes of the link typically is meant information on the parent or children object IDS that are associated since this is a graph like structure and a single parent can have multiple children and a child can also have multiple parent objects.

The operation of the invention is to take provided input in the form of an electronic document structured or unstructured or data streams which embodies textual and non textual content typically represented by graphical imagery such as pictures or illustrations and drawing data represented in the form of vectors or commonly known as descriptions on how to draw a line from position A to B. Electronic documents are best described but not limited to business or personal correspondence letters, product information and representation

thereof, parts manuals, corporate or personal financial information and such like. **Figures 15** and **16** represent respectively a unstructured document and a structured document.

Figure 15 illustrates how a typical unstructured document **360** might look like. By unstructured is meant that the document has minimal navigation information and structured elements such as headers, footers, tables, and font style use upon which the invention can directly extract such format and navigational information for the purpose of delivering and reconstructing partial document content back to the invention's client side data processing system **250**. The server side **10** data processing capabilities then have to be employed to construct such navigational and reconstructing information by analyzing the unstructured document data.

The analyzation is done by detecting changes in font size, font type, font style and font positioning. By analyzing this all textual content elements **362** will be marked as headers by the following criteria's: largest used font size =14, font style = Bold, font type = Arial. This is reinforced by a text paragraph definition **364**, which all have font size =10, no font style and font type = Times. By identifying the headers **362** based on this information the invention can construct a navigational aid in the form of a Table of Contents where none exists for the unstructured document. An embedded image **366** will have the image insertion point coordinates x, y as well as a width and height information all in pixels. The system **10** of the invention can then provide a link to the image **366** when rendering this on the client side application processing system **12** and element **368** which is marked out because of a font style change within a text paragraph **364** will also be marked as a header, but ranked with a lower importance then the other headers **362** as it is positioned within the text paragraph **364**. The element **368** will then be used by the server side **10** application processing when creating a navigational aide in form of a Table of Contents as the element **368** is deemed to have navigational significance. Likewise elements of a type **370** and **372** will be detected by the

invention's server side processing capabilities as the fonts positioning of these elements are in close proximity in addition to the usage of the font styles Bold and Italic and finally usage of a Tabulator (TAB) character for spacing. These all indicate that this section has significance to the document **360** and can be marked as a navigational aide point when construction a Table of Contents. Finally a page element **374** defines a page number description that is associated as a known navigational index reference for all the unstructured content residing on the documents page 1.

Figure 16 illustrates a structured document **380** that can be processed by the server side data processing system **10** with the purpose of identifying a document navigational structure in the form of a Table of Contents and defining content excerpts with the intent to reconstruct it on the client side processing application **12**. A Table of Contents (TOC) **382** is defined as an existing element in the structured document. Content elements **384** and **386** referenced as Heading 1 and Heading 2 are defined as paragraph headers **388** and are used as navigational elements in the construction of the TOC entity **382** by the document itself. The server side data processing application **10** can therefore directly extract these elements without the processing required for an unstructured document. Further the System **10** will be parsing the structured document to identify a hyperlink **390** to a bookmark **1**. A hyperlink is a navigational link between to objects or entities within a structured document indicating that they are connected and related and that by interacting with the hyperlink the user can navigate to a different part of the document. The system **10** will make a record of the hyperlink entity **390** and use it for navigational purposes. A bookmark entity **392** is another element defined within the structured document **380** itself that provides a navigational anchor point that the system **10** will make a record of for the purpose of allowing the execution of the navigational hyperlink entity **390**. Hyperlink entities **392, 394, 396** are different from the hyperlink entity **390** by being external hyperlinks. External hyperlinks are defined as link

objects pointing to other content or external applications or processes defined within the structured document **380** itself.

One hyperlink **392** will launch an Internet Browser with the address pointing to a website address www.arizan.com, or second hyperlink **394** to www.air2web.com, while the link **396** will launch an email application program with the "To" address set to sales@arizan.com. The server side data processing system **10** will, while processing the structured document, record these elements in the constructed Document Object Model for navigational or possible application launch purposes for software applications employing the invention. An Image element **398** in the structured document **380** provides information on an image/picture and position x, y and width, height information that the System **10** invention will record in the DOM thus giving the client component **12** the ability to reconstruct this element on a wireless device **12** in the right visual and content context. A table element **400** as defined by the structured document **380** contains content displayed and organized in a table like fashion. The invention's server side data processing functionality will read this information and map it when constructing the DOM utilizing the Document Engines table component to allow the invention's client side processing system to reconstruct this in a table like structure on a wireless device/pda **12**.

A graph element **402** in the structured document contains information in a graph type format such as bar chart and links to the data provided in the table element **400** used when drawing the chart by the application reading the structured document such as a word processor like MS Word by Microsoft Corporation. The server side data processing capabilities will retrieve and record the graphs link and rendering information when processing the structured document for the intent of having the client side data processing system reconstruct the graph on a wireless device/pda **12**. A header element **404** defines page header information for each individual page in the structured document, while a footer

element **406** defines page footer information that will be reused for each single page in the structured document **380**. The server side data processing capabilities will record the elements **404** and **406** and construct the appropriate entries in the DOM. A page entity **408** also provides a navigational index while navigating the structured document **380** as each page entity is a sequential entry that allows the user to look up an individual page within the page range starting with start page and ending with last page. The structured document as represented by **Figure 16** also embeds non-visual descriptive information about the document referred to as metadata **410**. The invention's server side data processing capabilities are able to extract this information and store it in the DOM where it can later be used by the system **10** to provide document information such as document name, author, date, time created etc. as defined by the metadata stored in the structured document.

Figure 17 illustrates the Document Object Model **420** as constructed by the invention's server side data processing system **10**. The illustration **420** represents the load document's general structure **422**. The DOM's entry point is a content node **424.** Each of a plurality of containers **426** can have multiple components of different types. The invention server side data processing application has four component types defined: a table component **430**, a text component **431**, an image component **432** and a vector component (not illustrated). Each component can also be referenced by other components under the same container **426** or different containers in the form of a reference command

Figure 18 illustrates the text component **430** as employed by the server side data processing system **10** after constructing the Document Object Model (DOM) **420** and after loading the structured document **380** illustrated by **Figure 16**. Each text component **430** can have more than one section command **440**, and as typically found for any standard structured or unstructured document, multiple paragraphs **442** can be included in one single section **440**. The paragraph object **442** has one attribute **444** defined, which is a reference to a Table Of

Content hyperlink. A hyperlink start object **446** itself is defined by its attribute object **448**, which describes the hyperlink ID and the destination of the link. Then a text segment object **450** has a describing attribute **452** that holds the textual content itself for the text object **450**. At the end of the paragraph object **442** a Hyperlink End object **454** is constructed with an attribute object **456** containing the Hyperlink pair ID.

Figure 19A illustrates a Document Object Model **460** of a loaded structured document **380** and the DOM composition for a paragraph as employed by the server side data processing system. A paragraph object **462** has an attribute object **464** that describes the associated style and the textual content of the paragraph **462**. A Font object **466** contains a font command and an associated attribute object **468** contains the font description and the font style information. The next command object linked to the paragraph object **462** is a Bookmark object **470** and it's descriptive attribute object **472**, which contains the Bookmark ID. Next the paragraph object **462** is linked to a text object **474** and it's associated attribute object **476** with the value of the textual content itself. **Figure 19B** then continues the DOM paragraph composition **460** started by **Figure 19A**. The next command object linked to the paragraph object **462** is a text command object **478** and an associated attribute object **480** containing the text value. Next is a new font command object **482** and an associated attribute object **484** containing the font type and style definition. Then the following objects linked to the paragraph object **462** are respectively text command and font command objects and related attributes **486 - 496** until the paragraph composition ends with a reference command object **498** and an associated attribute object **500** containing the reference value.

Figure 20 illustrates the table component **428** as employed by the server side data processing system **10** after constructing the Document Object Model (DOM) and after loading the structured document **380** illustrated by **Figure 16**. Each table component **428** can have multiple row objects **510, 512,** in which there might be one or more cells contained.

Figure 20 illustrates the first row of the sample table in the sample structured document **380**. The table row command object **510** links to a cell command object **514**, which has an associated attribute object **516** containing the textual content value of the cell. Additional cell command objects **518, 520, 522** and **524** will then be linked to the table row command object **510** until no more data is available for that row.

Figures 21A - 21F illustrate the server side data processing systems **10** description of content output in the form of a Universal Content Stream (UCS) as employed by the invention to transport data in a effective and optimized way for wireless delivery of the data over the Internet **16** to be processed by the client side data processing system **12**. **Figures 21A – 21F** has two tables each representing the server side processing capabilities in outputting a UCS data file representing commands to reproduce a structured document **380** and a unstructured document **360** as represented respectively in **Figures 16** and **15**.

Referring to **Figure 22**, a layout of one UCS format **530** is illustrated. The UCS (Universal Content Stream) format **530** provides the mechanism and data transport layer between the server **10** and the client technology or device **12**. The format primary purpose is to enable only the required visual image being sent in an optimal fashion over the available bandwidth. The format **530** supports the transfer of graphical images/pictures, 2D/3D drawings and various textual data such as word processor, and spreadsheet content.

The UCS can include several different standard formats for information transport in addition to the transcerpt command and data transfers to utilize the transcerpt technology and any available bandwidth. Embedding optional standard formats within UCS opens up for easy integration with client technology that do not require transcerption support. And for formats such as image/raster data the existing/coming standards such as JPEG 2000 will offer adequate support for compression without data loss from the server **10** to client **12**.

Examples of the industry standard formats that can be accessed by both the server **10** and other clients directly are the following four types:

- XML for textual, metadata and content information
- SVG for vector information
- JPEG 2000 for images
- OpenGL 3D representation

Other standard formats or proprietary formats can be easily added or removed through the plugable and scalable architecture of the component based server **10**.

The UCS format **530** includes three different data layers that together with optional industry standard formats make up the UCS format **530**. The three proprietary layers are: a TCL layer, a UCS textual content layer **534**, and a UCS image or vector data layer **536**.

The TCL (Transcerpt Command Language) layer **532** provides a data channel for two-way communication between the server **10** and the client **12**. The client **12** sends requests to the Client handler on the server side and receives a response with an transcerpt data stream in return. The TCL handles the following types of commands: Transcerpt requests for visual images, VQ (View and Query) commands, directory information, published document search/retrieve, and any required third party synchronization information. The TCL language itself is also constructed using XML for easy adaptation and future expansion.

The UCS vector stream **536** is a highly optimized vector format to provide vector-drawing capabilities on the thin client software. This enables the thin client software to offer vector representation for higher quality and accuracy on even small PDA's or handheld devices. The vector stream transfer and rendering is controlled through the TCL layer **532** which ensures that the Transcerpt handler will only generate a vector representation for the clipped viewable area requested by the client **12** and also to render it in an optimal fashion for

the limited client display area that generated the request. The UCS format **530** is designed in such a fashion that new proprietary layers or formats can easily be added. Any of the formats used for transportation of images, vector and text can also be replaced or added depending on integration requirements by third parties.

The Universal Content Stream (UCS) **530** is a binary file format specification, which enables delivery of original document content including, but not limited to textual content, image data, vector data and navigational information for the processed document. The UCS format is very efficient in overcoming the current limitation of the wireless devices **12** in respect to available bandwidth, which today in general is in the range of 9.6 to typically 48 although higher speeds are theoretically possible. The UCS addresses the limited bandwidth problem in a number of ways.

The UCS format **530** allows original document file sizes to be reduced by up to ninety percent (90%) while retaining all essential content and necessary format information for reconstructing the document on wireless devices using the client side processing capabilities. This reduction in size is achieved by stripping out non-essential formatting information that is used by the original document application, e.g. MS Word for rich presentation on desktop computer. Non-essential formatting information might be use of different font styles, font colors, and revision information, etc. which typically have no value on the wireless device **12** as the device screen might be monochrome or grayscale and the device typically has only one font type. The UCS file format also allows server side compression of the UCS file and client side decompression for further reduction in size. For compression common available technology in the form of compression algorithms such as LZW can be utilized. In addition the server side processing capabilities in producing transcerpts (excerpt of content for transmission) allows the generation of multiple UCS files for individual transmission from the server to the client in an iterative process where the resulting UCS file sent to the wireless

device is a result for a specific content query from the user. The iterative query process from the user on a wireless device requesting content from the server might typically result in the delivery of for example ten individual UCS files delivered to the device. The UCS file format includes information on how these individual UCS files link together so that the original document can then be reconstituted by the individual files in memory for a coherent viewing or reading experience. The UCS format also contains the Document Object Model (DOM) ID to ensure that the server **10** processing capabilities can delivery the correct DOM information to the requesting device **12**. In addition the UCS file format also contains Internet hyperlinks allowing the client to request more content from within the document. Command codes can also be embedded in the UCS format to facilitate the execution of specific processes by the server side data processing capabilities. This is typically a request for specific content, lookup in the navigation information stored within the DOM, etc.

DEFINITION OF TERMS

1. Anatomization – A specific technology process for parsing and analyzing a structured or unstructured document or data source with the purpose of establishing a uniform Document Object Model containing all content and a navigation document hierarchy.

2. Bookmarks – A navigational link associating a navigational reference from position X to position & within a document.

3. DOM – Document Object Model (DOM), a programming interface specification being developed by the World Wide Web Consortium, defines how to create and structure XML based documents as objects.

4. External File – reference to an associated external data or document file outside the original document.

5. External Object – reference to an associated external data object outside the original document.

6. Font Color – The color of the font.

7. Font position – The rendering x, y pixel or character positioning information.

8. Font Size – Font size in points indicating the size of a font character.

9. Font Style – The style of the font as in normal, italics, bold, underline, etc.

10. Font Type – Name of font type used in document e.g., Times Roman.

11. Footer – Footnote at the bottom of an individual or all pages in a document containing content.

12. Header – Header at the top of an individual or all pages in a document containing content.

13. Hyperlinks – A navigational link external or internal to the document content referencing a bookmark or external data or content source.

14. Page – Defines an individual age in a document containing content information typically separated by a page break character of a fixed page or printer page size definition.

15. Paragraph – Collection of textual characters or lines of text grouped together in a cohesive body separated by line breaks.

16. Predictive Caching – A specific technology process designed to cache relevant or associated content on the server prior to required delivery/transmittal to a wireless/ wired device to minimize bandwidth and network latency.

17. Table of Content (TOC) – Presents a hierarchical navigational structure to the document content for informing and assisting in navigation of the document content.

18. Title Styles – Predefined styles in a document centric creation application to allow for the construction of a Table of Content such as Heading 1, Heading 2, Heading 3, etc.

19. Transcerption – A specific technology process for delivering only the requested content excerpt and transmitting this to the client.

20. USC – The Universal Content Stream is a content data format optimized for minimized data size while preserving content context, navigational and presentation information for delivery to wireless/wired devices.

21. XML – extensible Markup Language is a flexible way to create common information formats and share both the format and the data on the World Wide Web, intranets, and elsewhere.

CLAIMS

1. A method for delivering content to a portable computing device, comprising:

receiving a request from the portable computing device for content associated with a document remote to the portable computing device;

identifying one or more portions of the document associated with the requested content;

reconstructing the identified portion or portions of the document from elements of the document stored in a Document Object Model (DOM) Storage; and

transmitting the reconstructed identified portion or portions to the portable computing device for display thereon.

2. The method of Claim 1, wherein identifying one or more portions of the document includes utilizing a lookup or navigation tree to identify one or more portions of the document.

3. The method of Claim 1, wherein identifying one or more portions of the document includes utilizing one or more probability algorithms to determine what are the most relevant portions of the document to the requested content.

4. The method of Claim 1, wherein reconstituting the identified portion or portions includes reconstructing the identified portion or portions in a predetermined format compatible with the portable computing device.

5. A method for publishing a document content on a remote portable computing device, comprising:

receiving the document to be published, wherein the document includes visual document representation and document content;

separating the visual document representation and the document content;

indexing the document content; and

establishing navigational links, which are mapped to a navigational tree; and storing the document content as Document Object Model structures.

6. The method of Claim 5, further including generating a metadata header for the document.

7. A method for wireless delivering document content to a portable computing device, comprising.

analyzing the content of a structured or unstructured document by identifying existing navigational elements or where none exist by analytical processing based on the documents visual representation to establish navigational links for the document content;

receiving a request from the portable computing device for content in said document;

using said navigational links to find and transmit said requested content; and

receiving said content at said portable computing device and reconstructing said content on said device.

8. The method of claim 7, including creating a table of contents during said analysis and said portable computing device utilizing said table of contents to request said content.

9. The method of Claim 7, including storing said content in a document object model and obtaining said requested content from said document object model.

10. The method of Claim 7, including transmitting said content in a universal content stream to said portable computing device.

11. The method of Claim 7, wherein analyzing said document includes anatomizing said document.

12. The method of Claim 7, including transmitting content excerpts of said document to said portable computing device.

13. The method of Claim 12, including predictive caching of said content excerpts relating to said content request and said content excerpt previously sent.

14. A system for wireless document content delivery to a portable computing device, comprising:

a communication bus having an input and output coupled thereto.

a document processing module coupled to said bus and to a document data base;

a preprocessing module coupled to said bus;

an anatomizing module coupled to said bus;

an indexing module coupled to said bus and to an indexing database; and

a transcerpt module coupled to said bus and to a transcerpt cache.

15. The system of Claim 14, including a predictive cache coupled to said transcerpt module and to said bus.

Figure 1

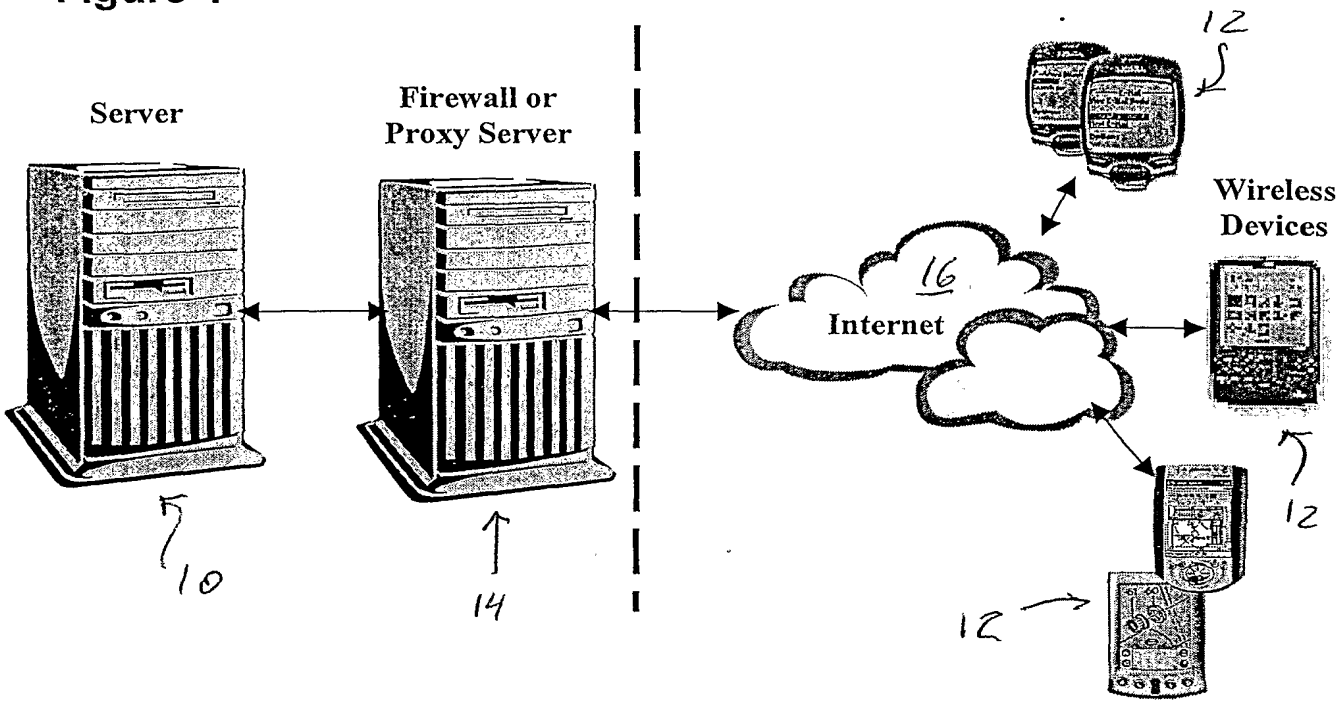

Server
Firewall or
Proxy Server
Internet
Wireless
Devices
10
14
16
12
12
12

Figure 2

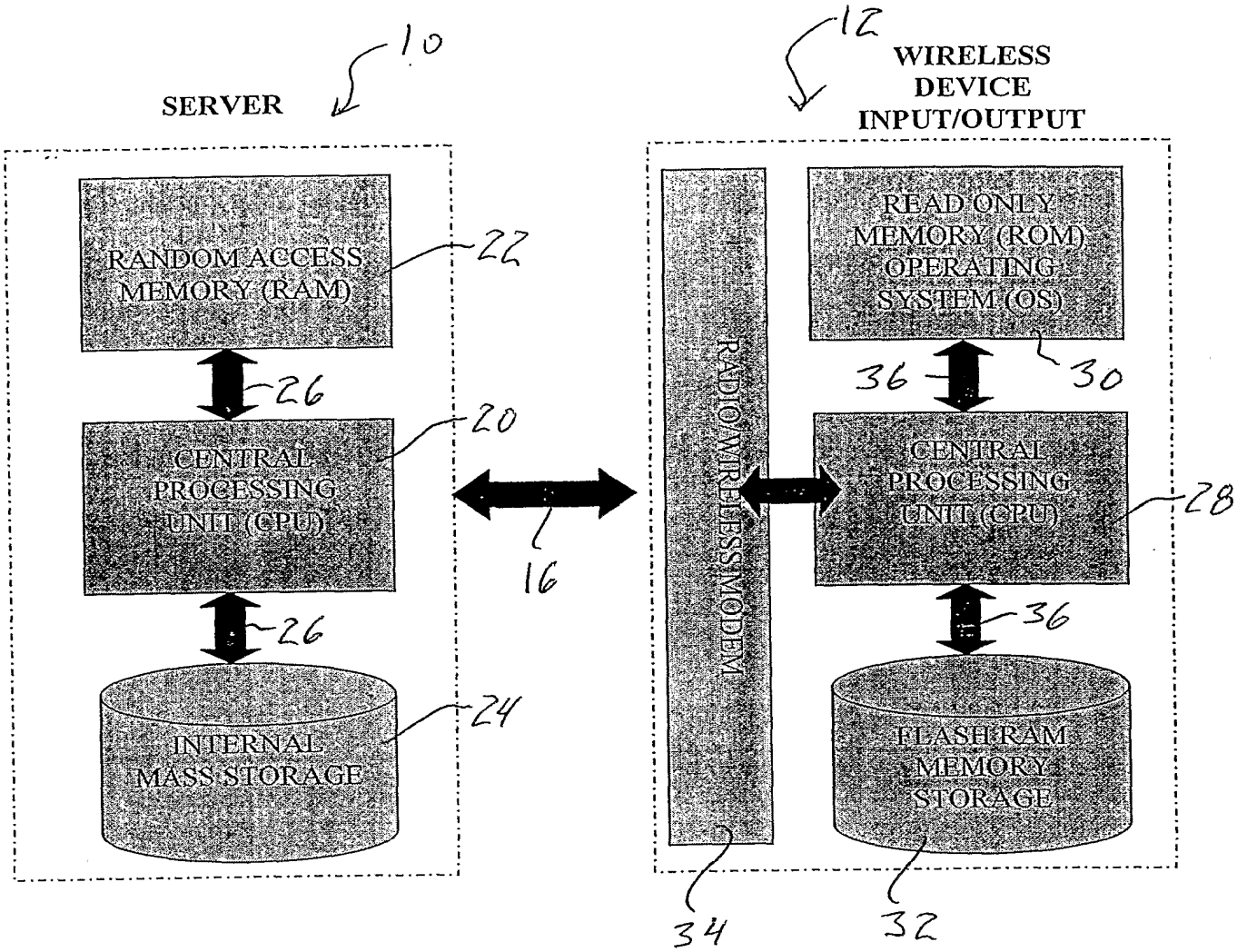

10
SERVER
12
WIRELESS DEVICE INPUT/OUTPUT
RANDOM ACCESS MEMORY (RAM)
22
26
CENTRAL PROCESSING UNIT (CPU)
20
26
INTERNAL MASS STORAGE
24
16
RADIO/WIRELESS MODEM
READ ONLY MEMORY (ROM) OPERATING SYSTEM (OS)
36
30
CENTRAL PROCESSING UNIT (CPU)
28
36
FLASH RAM MEMORY STORAGE
34
32

Figure 3

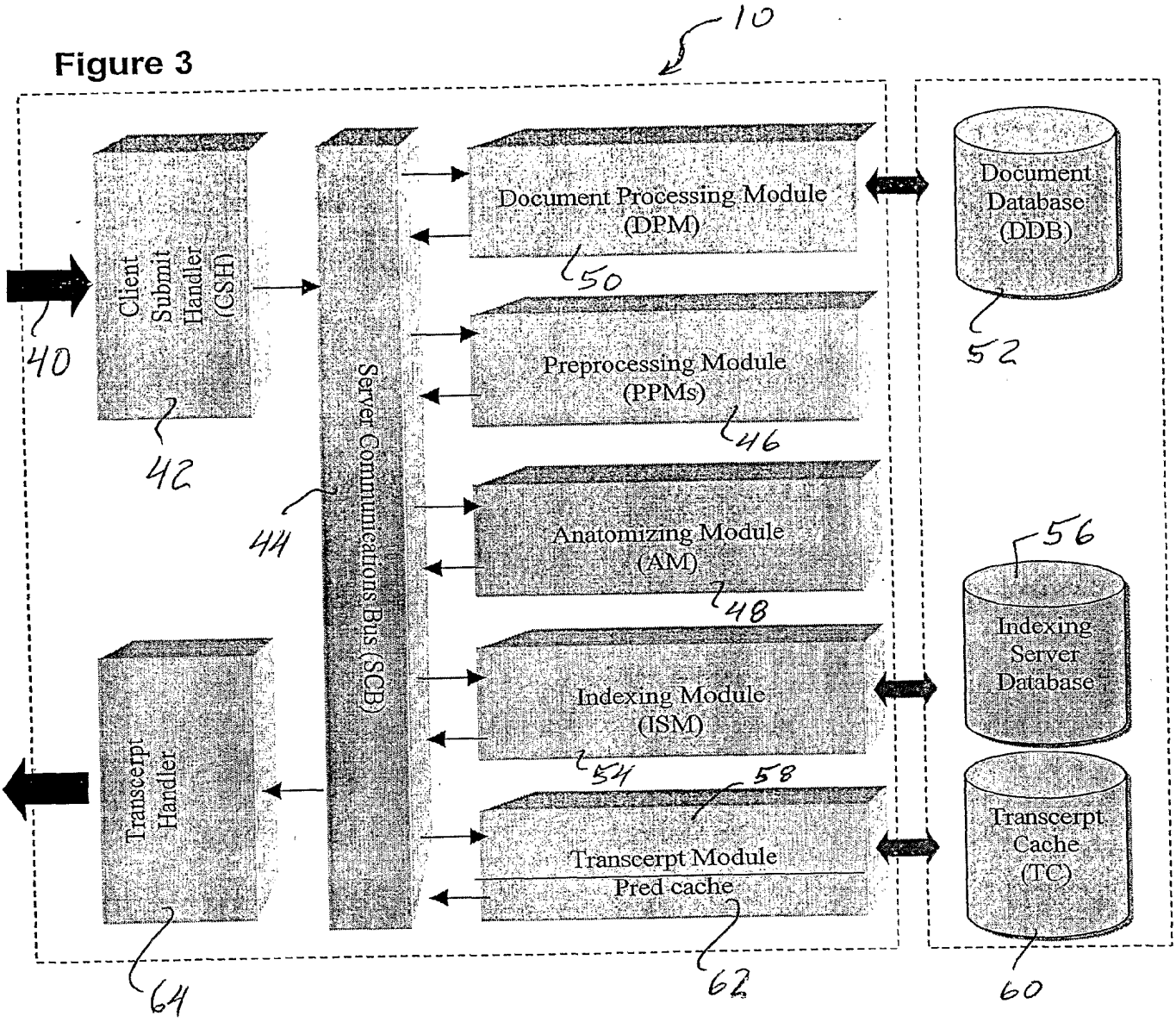

10
Client Submit Handler (CSH)
40
42
Server Communications Bus (SCB)
44
Document Processing Module (DPM)
50
Preprocessing Module (PPMs)
46
Anatomizing Module (AM)
48
Indexing Module (ISM)
54
Transcerpt Module
Pred cache
58
62
Transcerpt Handler
64
Document Database (DDB)
52
56
Indexing Server Database
Transcerpt Cache (TC)
60

Figure 4

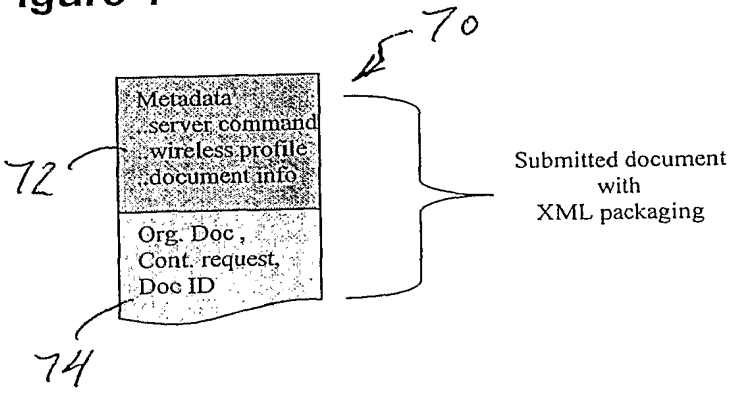

70
Metadata
..server command
..wireless profile
..document info
72
Org. Doc ,
Cont. request,
Doc ID
74
Submitted document
with
XML packaging

Figure 5

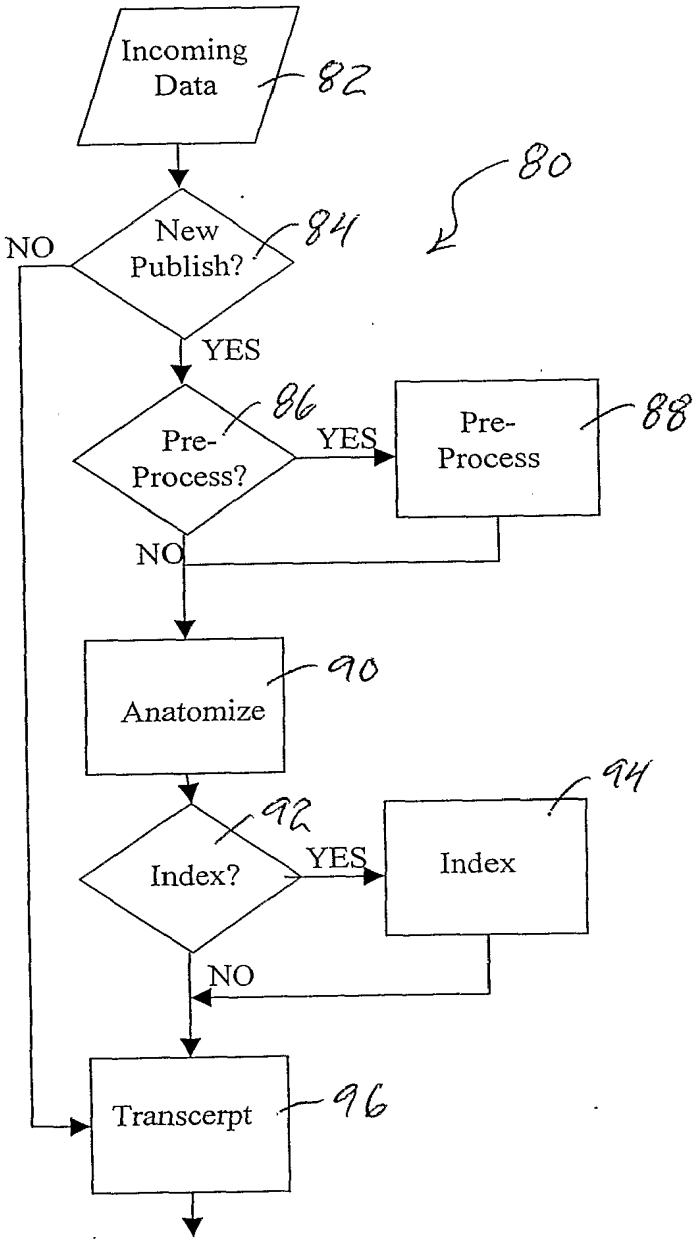

Incoming Data
82
80
New Publish?
84
NO
YES
Pre-Process?
86
YES
Pre-Process
88
NO
Anatomize
90
Index?
92
YES
Index
94
NO
Transcerpt
96

Figure 6

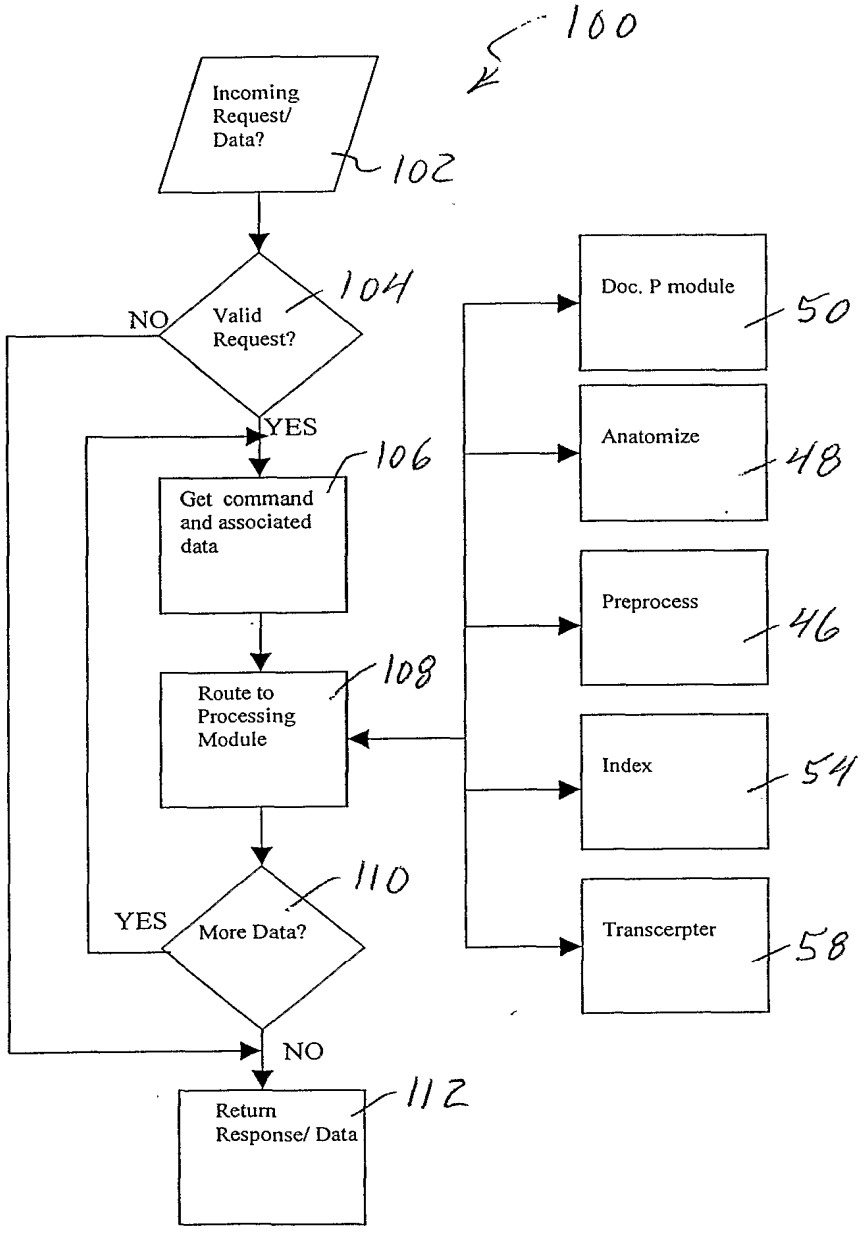

100
Incoming Request/ Data?
102
Valid Request?
104
NO
YES
Get command and associated data
106
Route to Processing Module
108
More Data?
110
YES
NO
Return Response/ Data
112
Doc. P module
50
Anatomize
48
Preprocess
46
Index
54
Transcerpter
58

Figure 7

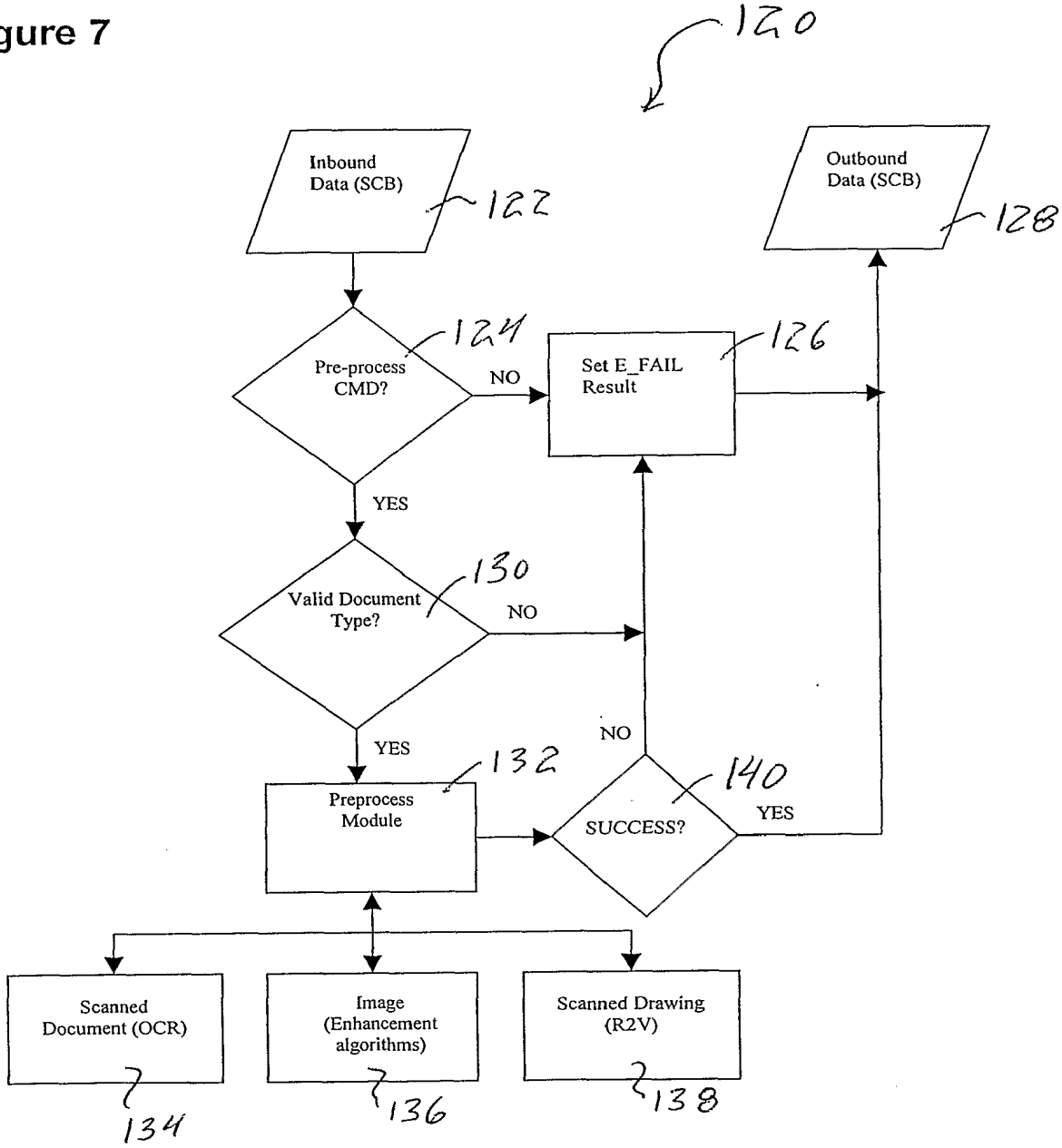

120
Inbound Data (SCB)
122
Outbound Data (SCB)
128
Pre-process CMD?
124
NO
Set E_FAIL Result
126
YES
Valid Document Type?
130
NO
YES
NO
Preprocess Module
132
SUCCESS?
140
YES
Scanned Document (OCR)
134
Image (Enhancement algorithms)
136
Scanned Drawing (R2V)
138

Figure 8

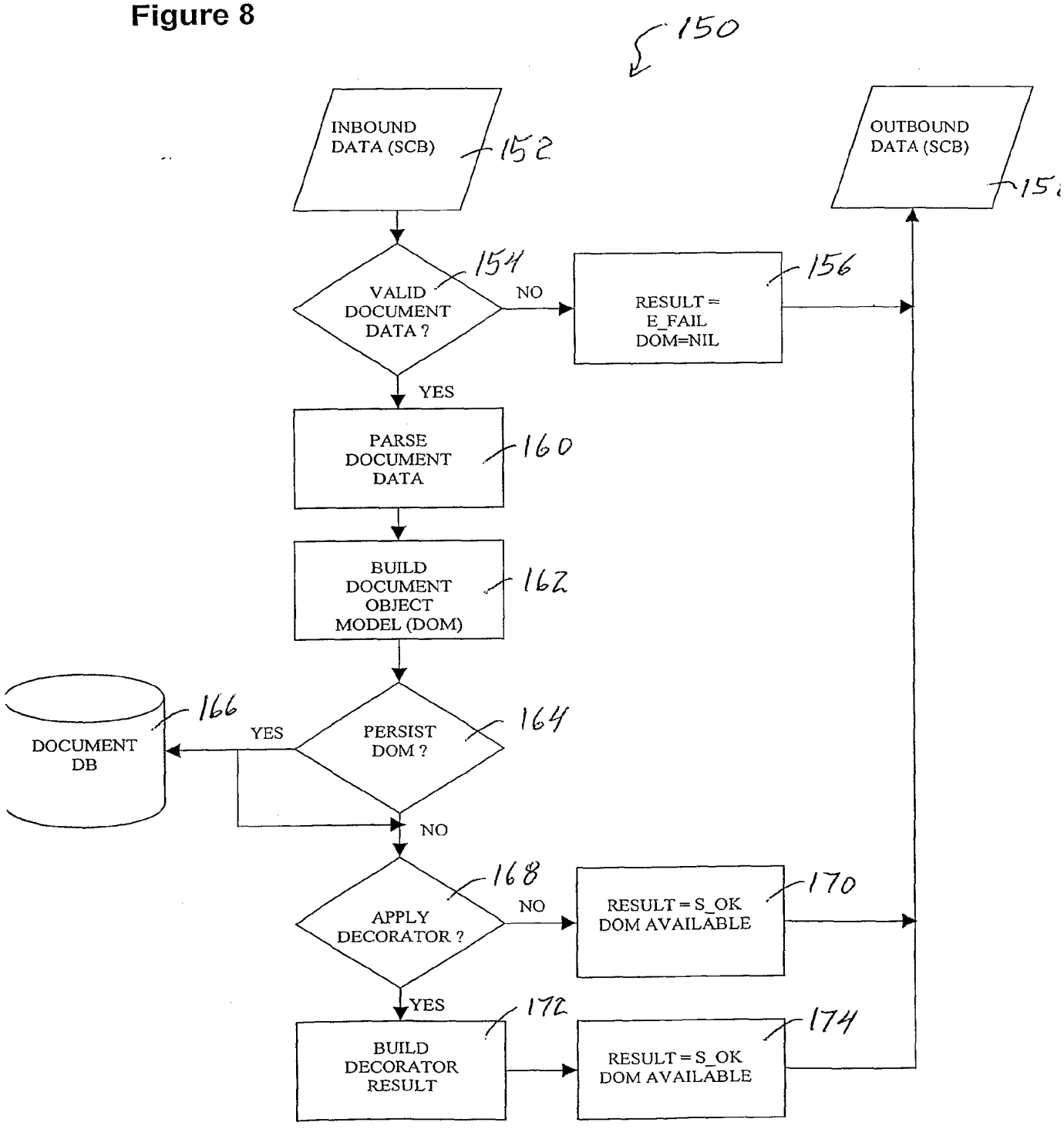

150
INBOUND DATA (SCB)
152
OUTBOUND DATA (SCB)
VALID DOCUMENT DATA ?
154
NO
RESULT = E_FAIL DOM=NIL
156
YES
PARSE DOCUMENT DATA
160
BUILD DOCUMENT OBJECT MODEL (DOM)
162
DOCUMENT DB
166
YES
PERSIST DOM ?
164
NO
APPLY DECORATOR ?
168
NO
RESULT = S_OK DOM AVAILABLE
170
YES
BUILD DECORATOR RESULT
172
RESULT = S_OK DOM AVAILABLE
174

Figure 9

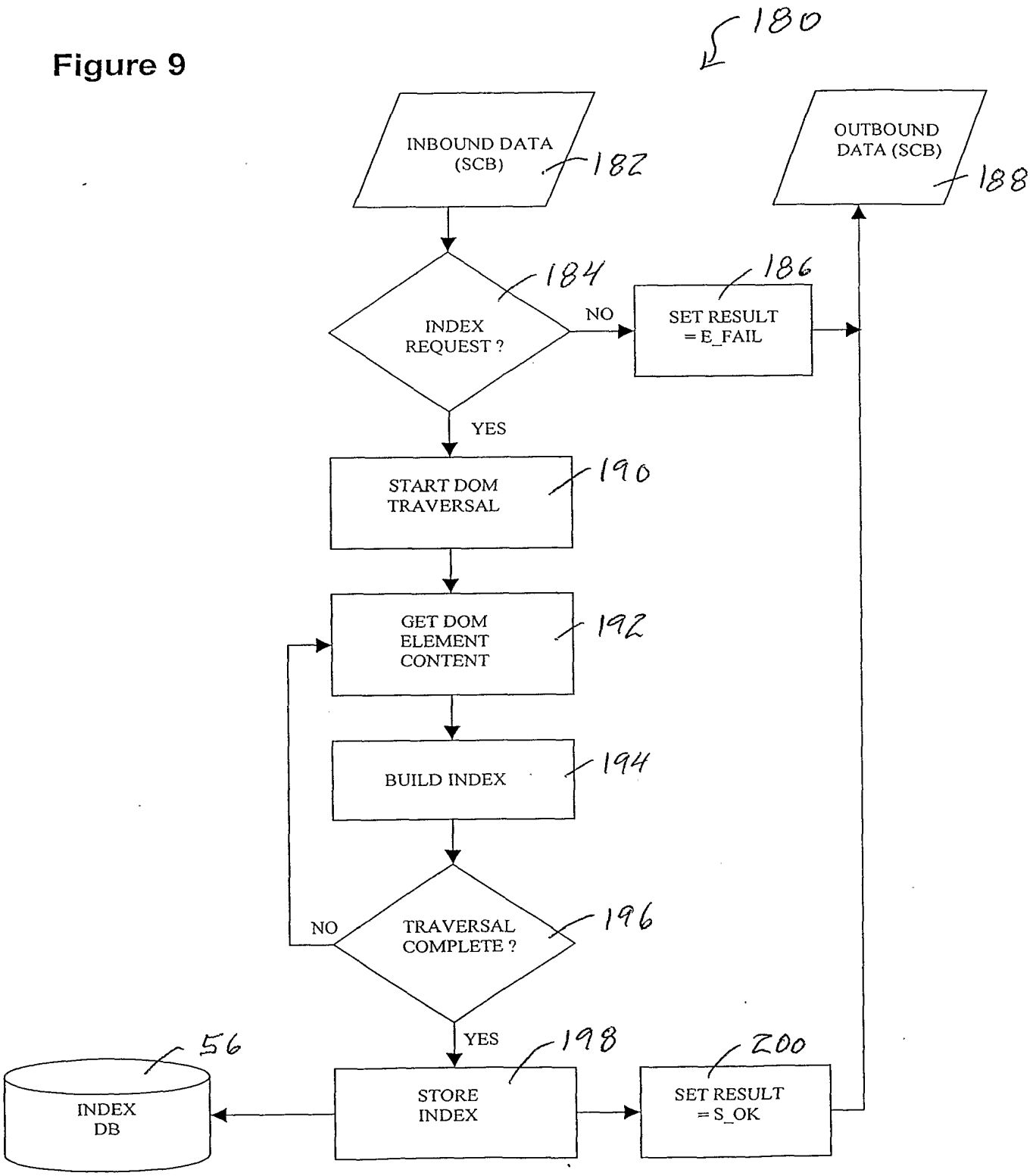

180
INBOUND DATA (SCB)
182
OUTBOUND DATA (SCB)
188
INDEX REQUEST ?
184
NO
SET RESULT = E_FAIL
186
YES
START DOM TRAVERSAL
190
GET DOM ELEMENT CONTENT
192
BUILD INDEX
194
NO
TRAVERSAL COMPLETE ?
196
YES
56
INDEX DB
STORE INDEX
198
SET RESULT = S_OK
200

Figure 10

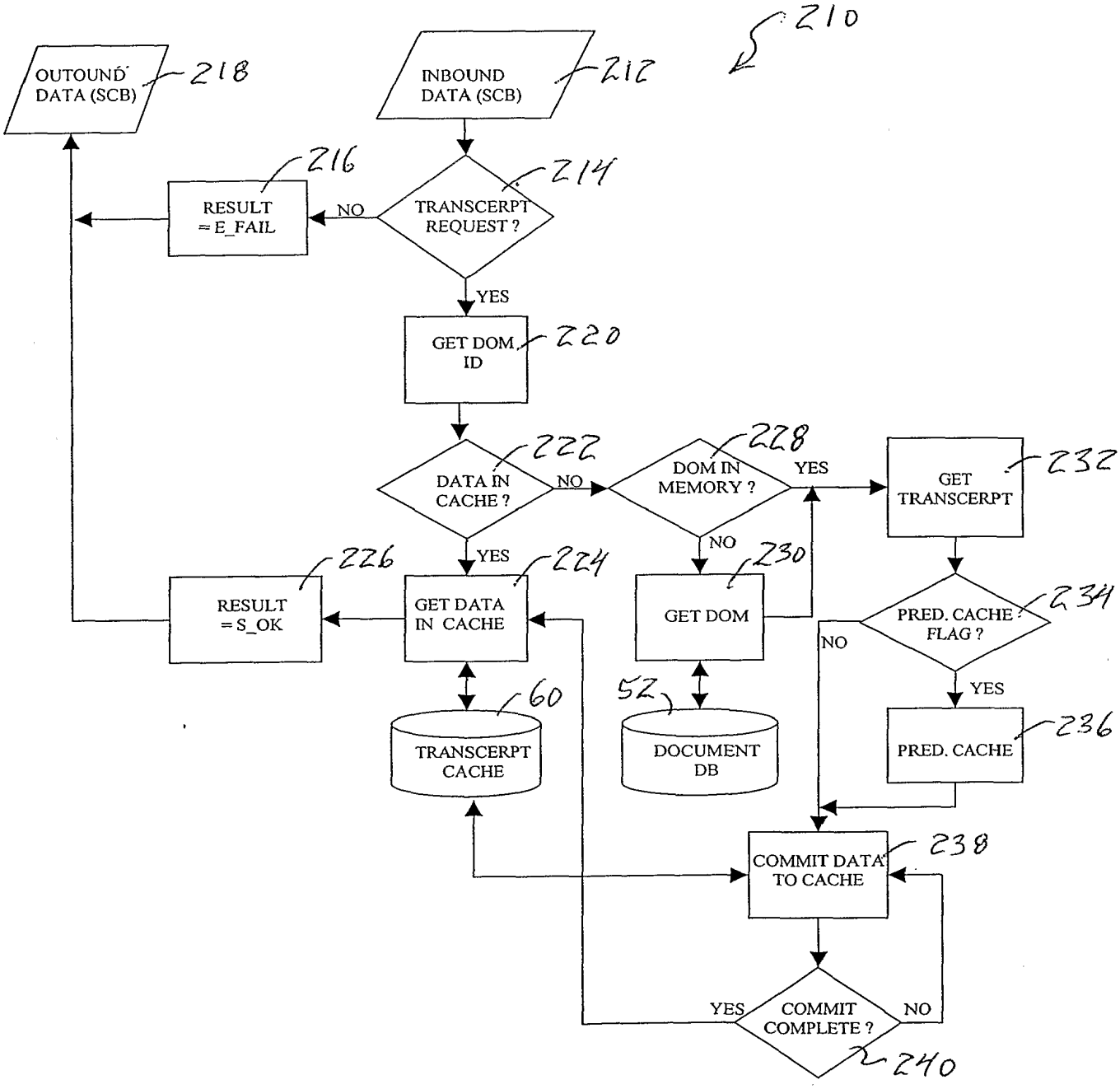

210
OUTOUND DATA (SCB)
218
INBOUND DATA (SCB)
212
RESULT = E_FAIL
216
NO
TRANSCERPT REQUEST ?
214
YES
GET DOM ID
220
DATA IN CACHE ?
222
NO
DOM IN MEMORY ?
228
YES
GET TRANSCERPT
232
YES
RESULT = S_OK
226
GET DATA IN CACHE
224
NO
GET DOM
230
PRED. CACHE FLAG ?
234
NO
YES
60
TRANSCERPT CACHE
52
DOCUMENT DB
PRED. CACHE
236
COMMIT DATA TO CACHE
238
YES
COMMIT COMPLETE ?
NO
240

Figure 11

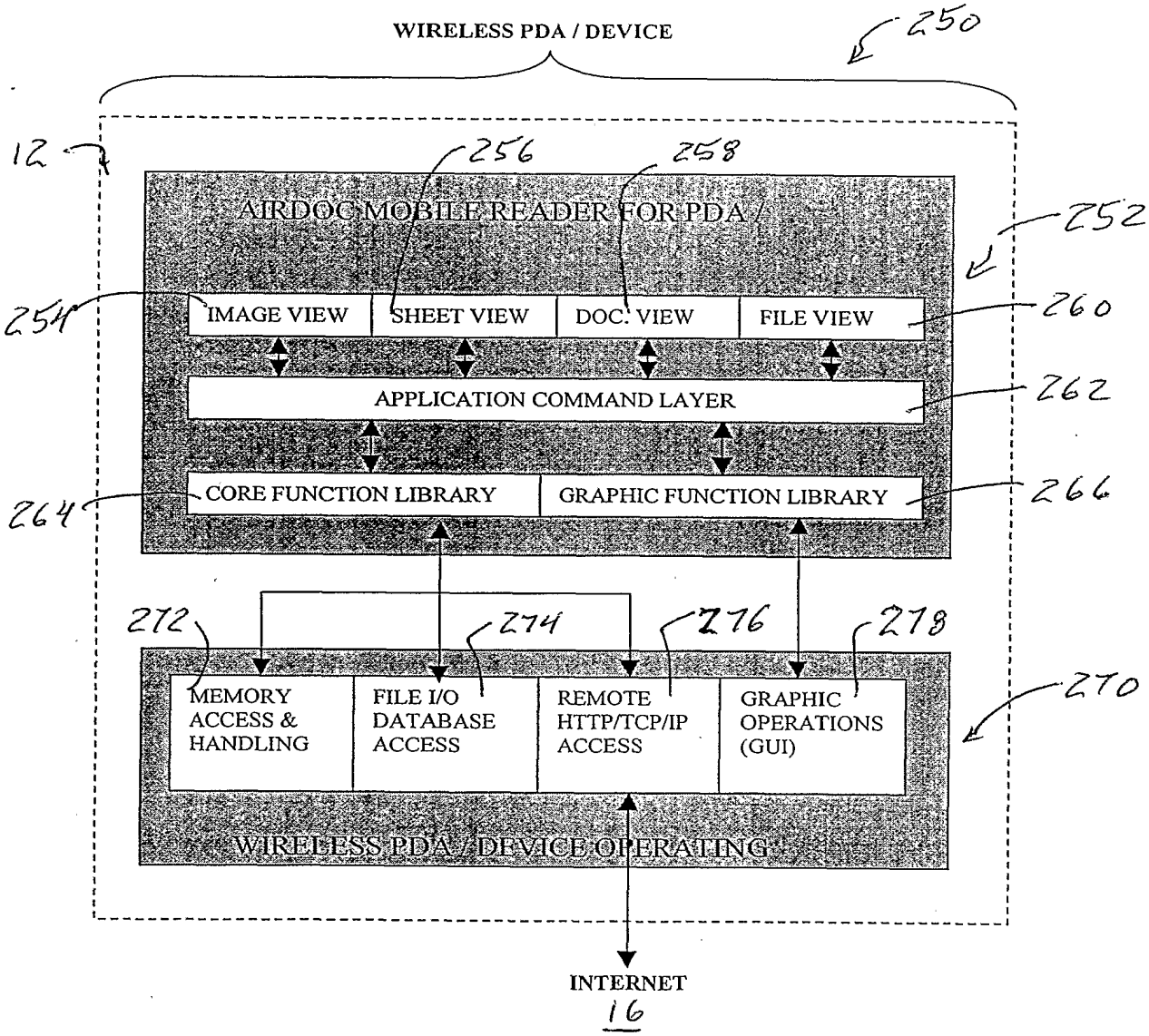

WIRELESS PDA / DEVICE
250
12
256
258
AIRDOC MOBILE READER FOR PDA
252
254
IMAGE VIEW
SHEET VIEW
DOC. VIEW
FILE VIEW
260
APPLICATION COMMAND LAYER
262
CORE FUNCTION LIBRARY
GRAPHIC FUNCTION LIBRARY
264
266
272
274
276
278
MEMORY ACCESS & HANDLING
FILE I/O DATABASE ACCESS
REMOTE HTTP/TCP/IP ACCESS
GRAPHIC OPERATIONS (GUI)
270
WIRELESS PDA / DEVICE OPERATING
INTERNET
16

Figure 12

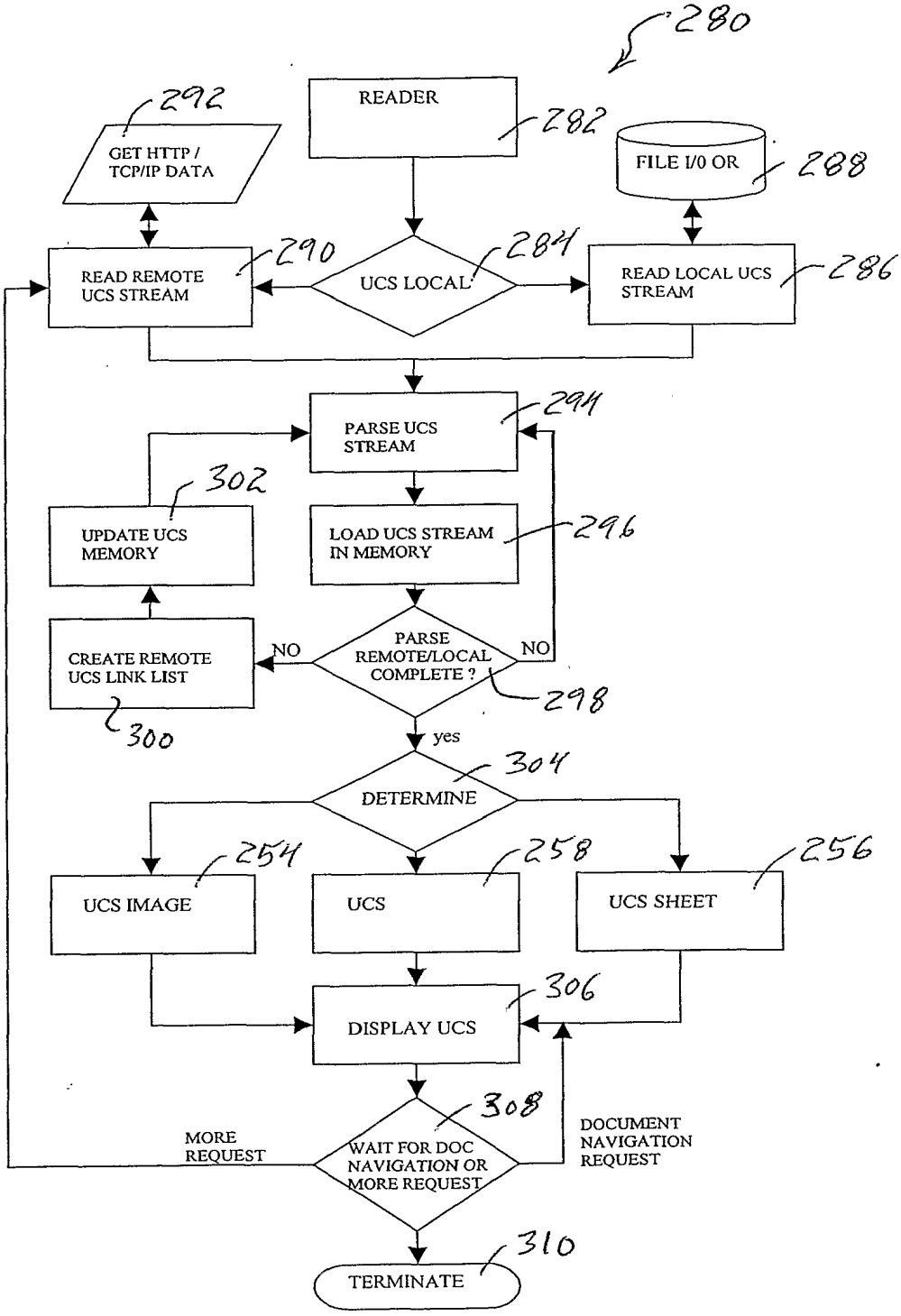

280
READER
282
GET HTTP / TCP/IP DATA
292
FILE I/0 OR
288
READ REMOTE UCS STREAM
290
UCS LOCAL
284
READ LOCAL UCS STREAM
286
PARSE UCS STREAM
294
UPDATE UCS MEMORY
302
LOAD UCS STREAM IN MEMORY
296
CREATE REMOTE UCS LINK LIST
300
NO
PARSE REMOTE/LOCAL COMPLETE ?
NO
298
yes
DETERMINE
304
UCS IMAGE
254
UCS
258
UCS SHEET
256
DISPLAY UCS
306
MORE REQUEST
WAIT FOR DOC NAVIGATION OR MORE REQUEST
308
DOCUMENT NAVIGATION REQUEST
TERMINATE
310

Figure 13

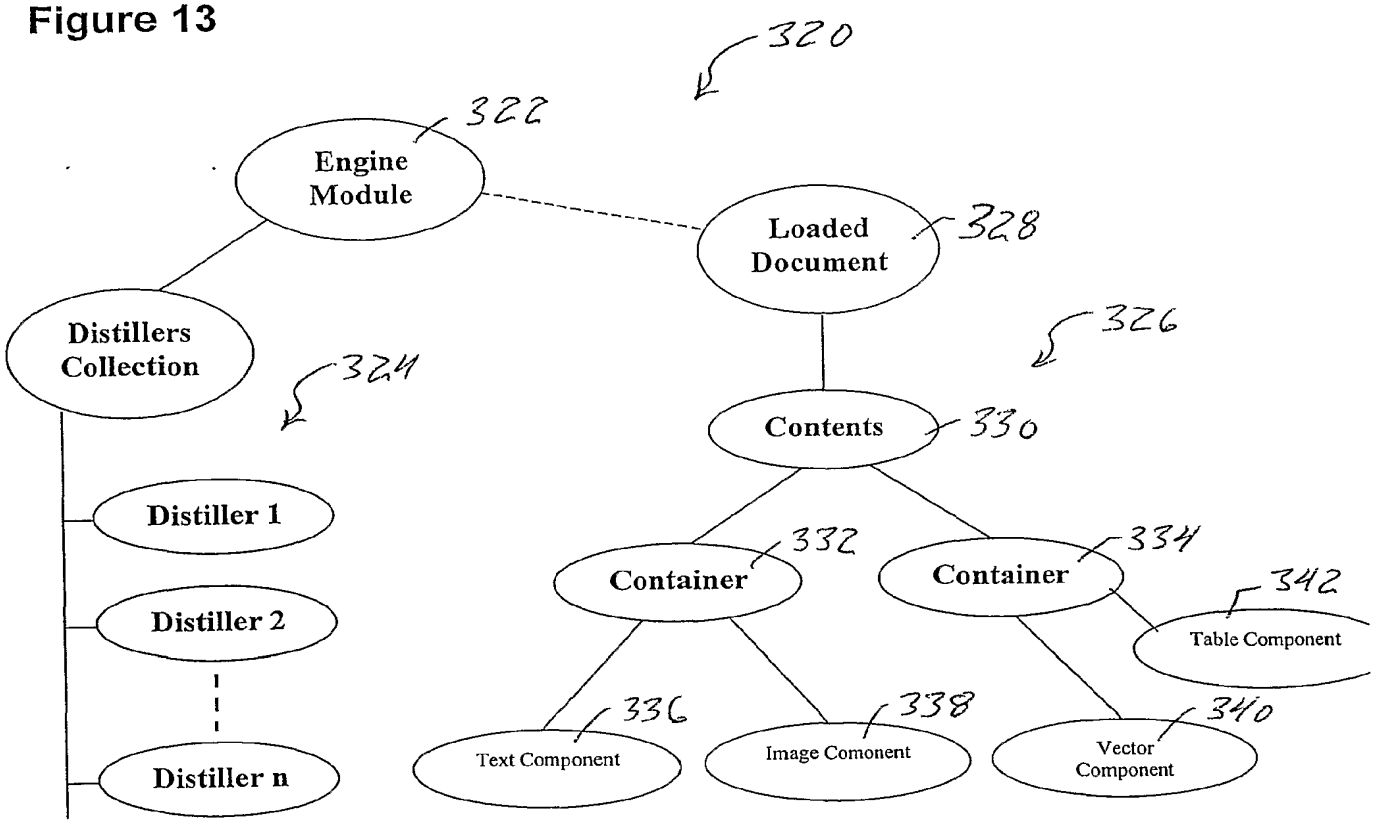

320
322
Engine Module
Loaded Document
328
Distillers Collection
326
324
Contents
330
Distiller 1
Distiller 2
Distiller n
Container
332
Container
334
342
Table Component
336
Text Component
338
Image Comonent
340
Vector Component

Figure 14

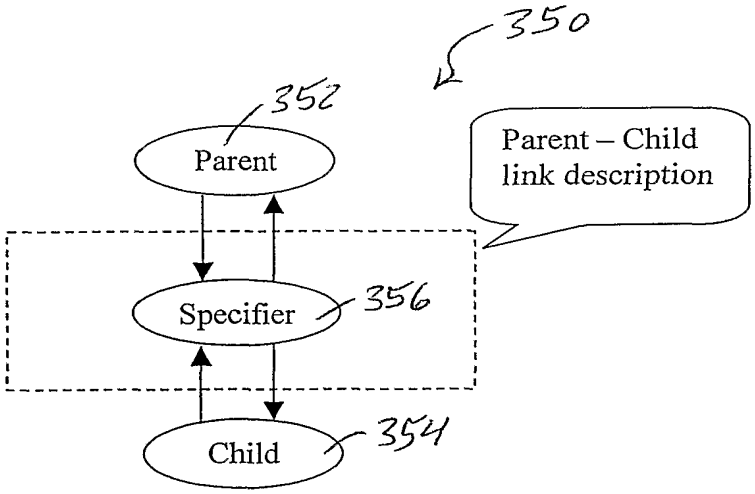

350
352
Parent
Parent – Child
link description
Specifier
356
Child
354

Figure 15

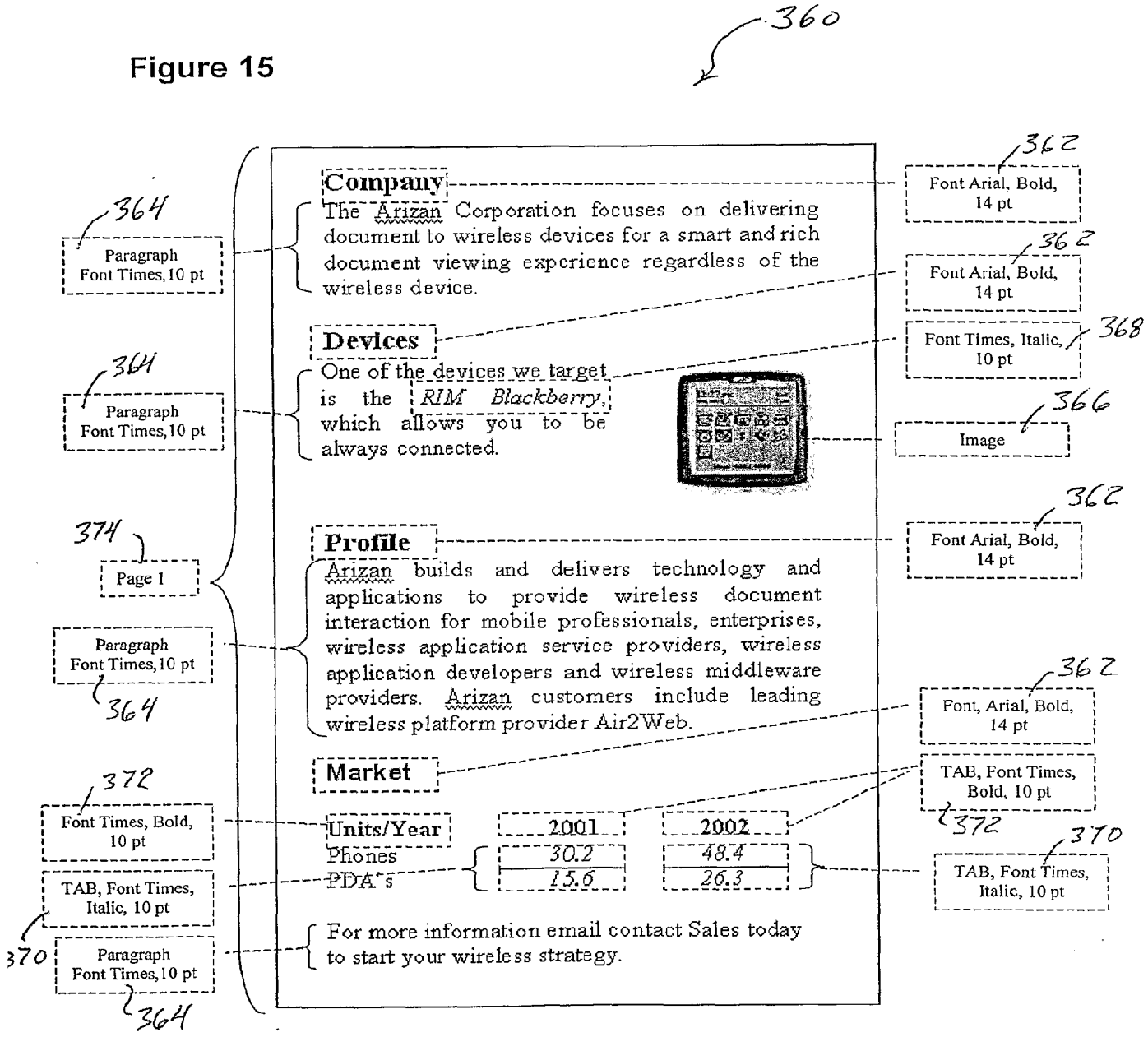

360
Company
The Arizan Corporation focuses on delivering document to wireless devices for a smart and rich document viewing experience regardless of the wireless device.
Devices
One of the devices we target is the RIM Blackberry, which allows you to be always connected.
Profile
Arizan builds and delivers technology and applications to provide wireless document interaction for mobile professionals, enterprises, wireless application service providers, wireless application developers and wireless middleware providers. Arizan customers include leading wireless platform provider Air2Web.
Market
Units/Year
2001
2002
Phones
30.2
48.4
PDA's
15.6
26.3
For more information email contact Sales today to start your wireless strategy.
362
Font Arial, Bold, 14 pt
364
Paragraph Font Times, 10 pt
368
Font Times, Italic, 10 pt
366
Image
374
Page 1
372
Font Times, Bold, 10 pt
TAB, Font Times, Bold, 10 pt
TAB, Font Times, Italic, 10 pt
370
Paragraph Font Times, 10 pt

Figure 16

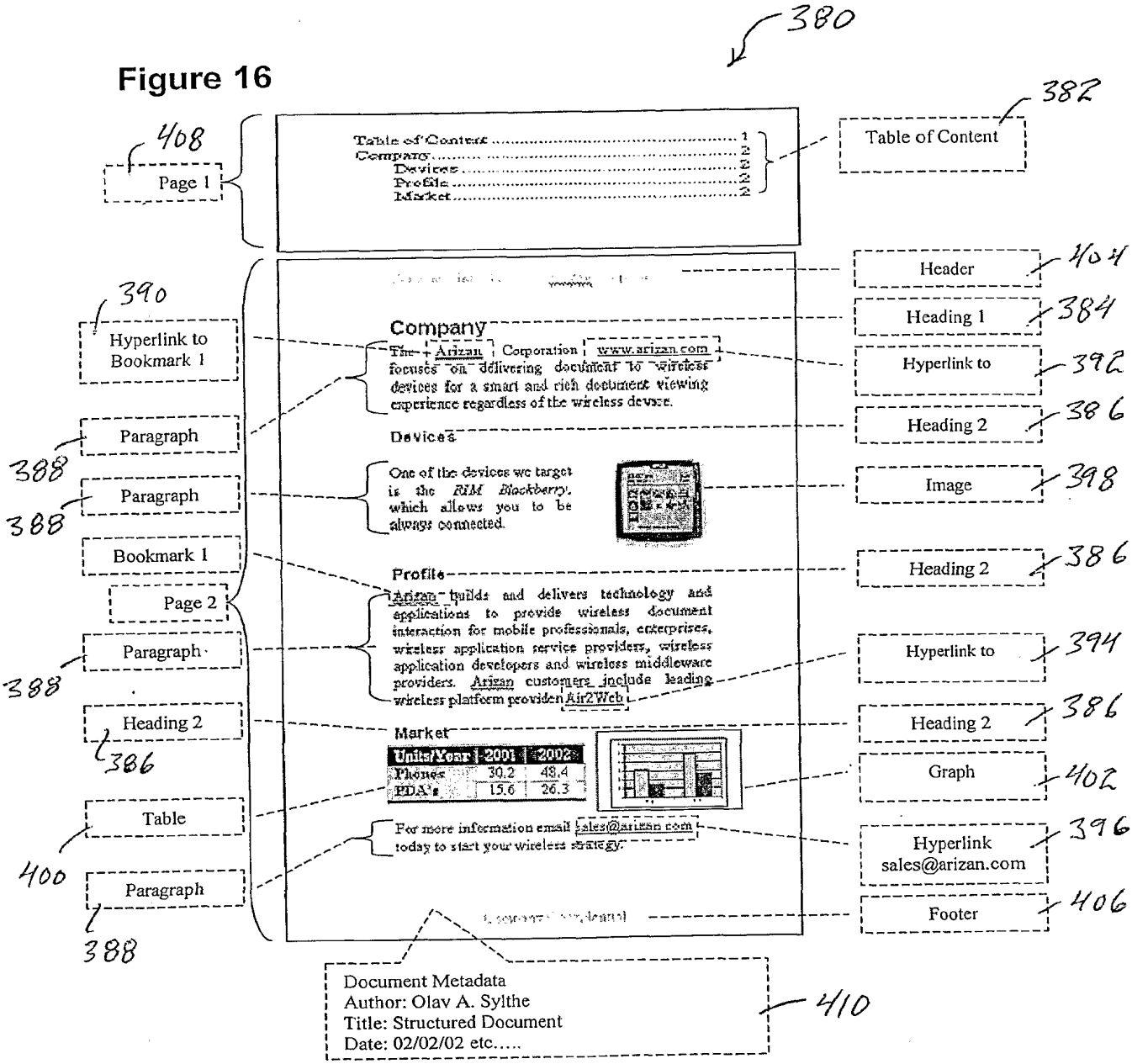

380
382
Table of Content
Table of Contents 1
Company 2
Devices 2
Profile 2
Market 2
408
Page 1
Header
404
390
Hyperlink to Bookmark 1
Heading 1
384
Company
The Arizan Corporation www.arizan.com focuses on delivering document to wireless devices for a smart and rich document viewing experience regardless of the wireless device.
Hyperlink to
392
Heading 2
386
Devices
388
Paragraph
One of the devices we target is the RIM Blackberry, which allows you to be always connected.
Image
398
388
Paragraph
Bookmark 1
Heading 2
386
Profile
Page 2
Arizan builds and delivers technology and applications to provide wireless document interaction for mobile professionals, enterprises, wireless application service providers, wireless application developers and wireless middleware providers. Arizan customers include leading wireless platform provider Air2Web.
Hyperlink to
394
388
Paragraph
Heading 2
386
Market
Graph
402
386
Table
Units/Year 2001 2002
Phones 30.2 48.4
PDA's 15.6 26.3
400
For more information email sales@arizan.com today to start your wireless strategy.
Hyperlink sales@arizan.com
396
Paragraph
388
Footer
406
Document Metadata
Author: Olav A. Sylthe
Title: Structured Document
Date: 02/02/02 etc.....
410

Figure 17
document

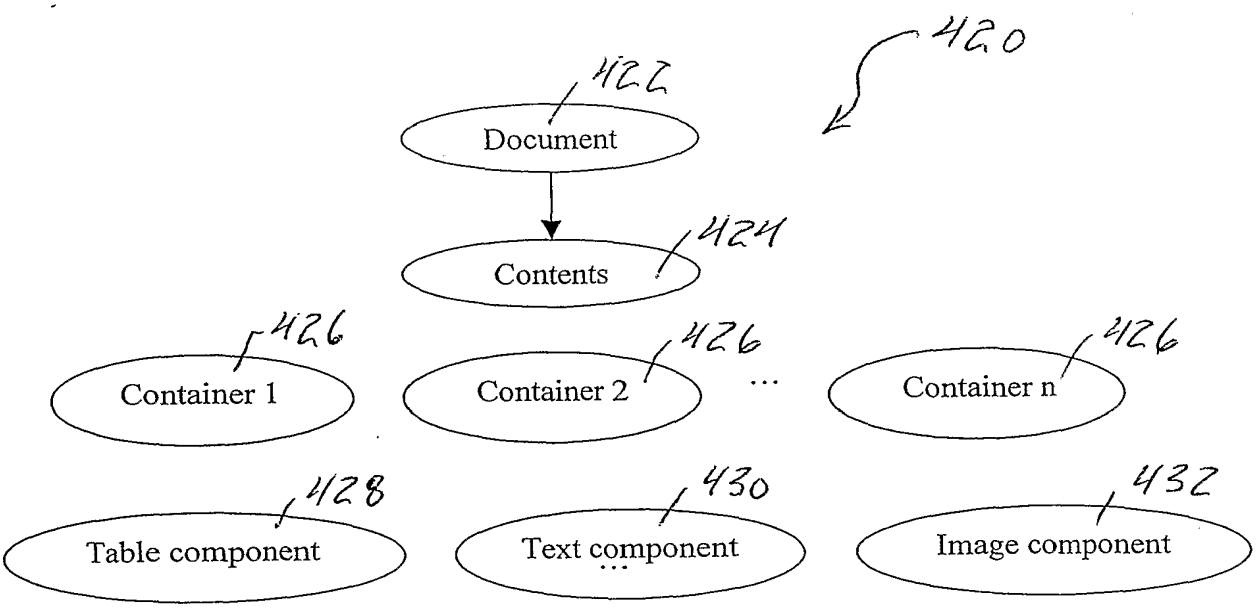


Figure 18

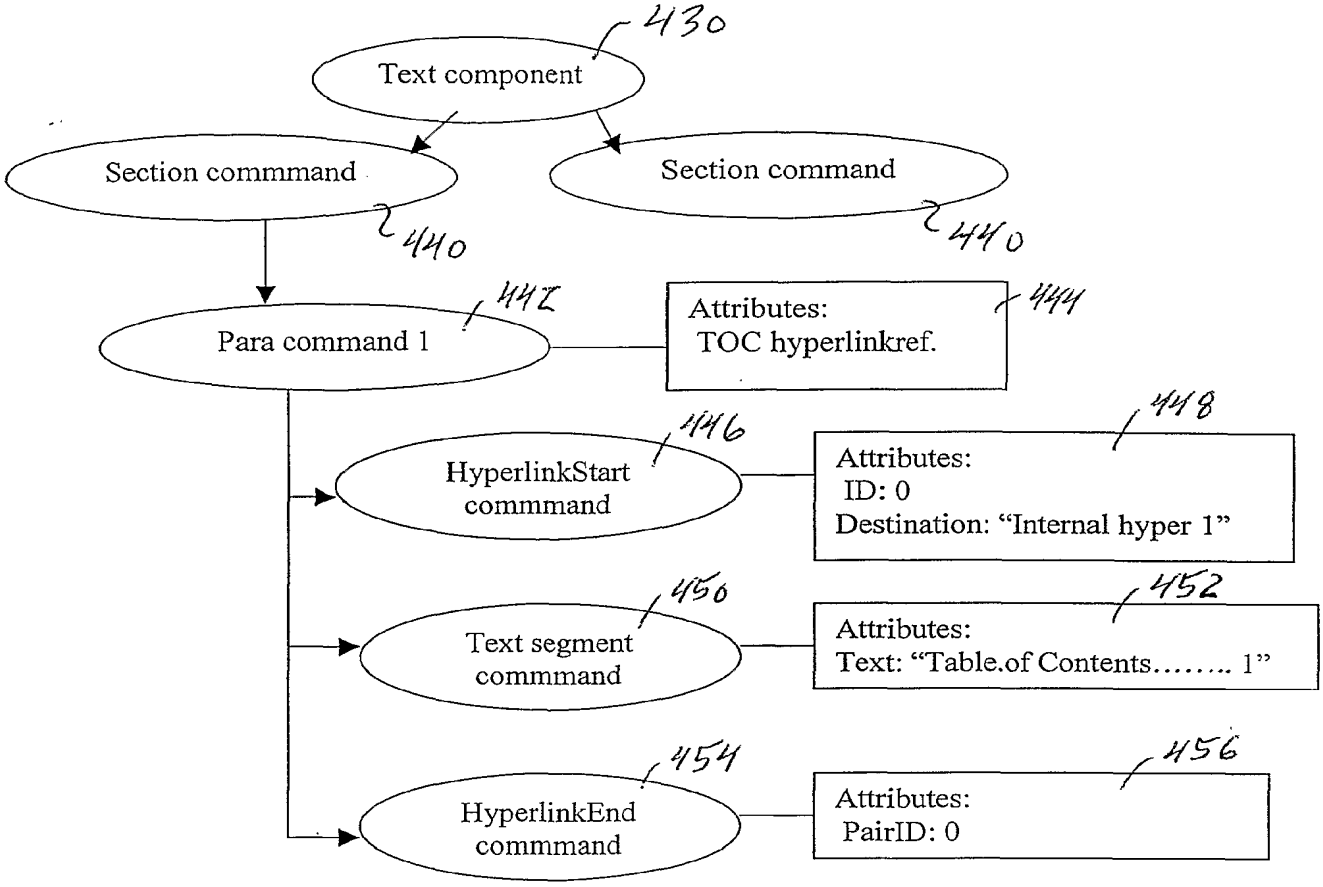

430
Text component
Section commmand
Section command
440
440
442
Para command 1
Attributes:
TOC hyperlinkref.
444
446
HyperlinkStart
commmand
448
Attributes:
ID: 0
Destination: "Internal hyper 1"
450
Text segment
commmand
452
Attributes:
Text: "Table.of Contents..... 1"
454
HyperlinkEnd
commmand
456
Attributes:
PairID: 0

Figure 19 A

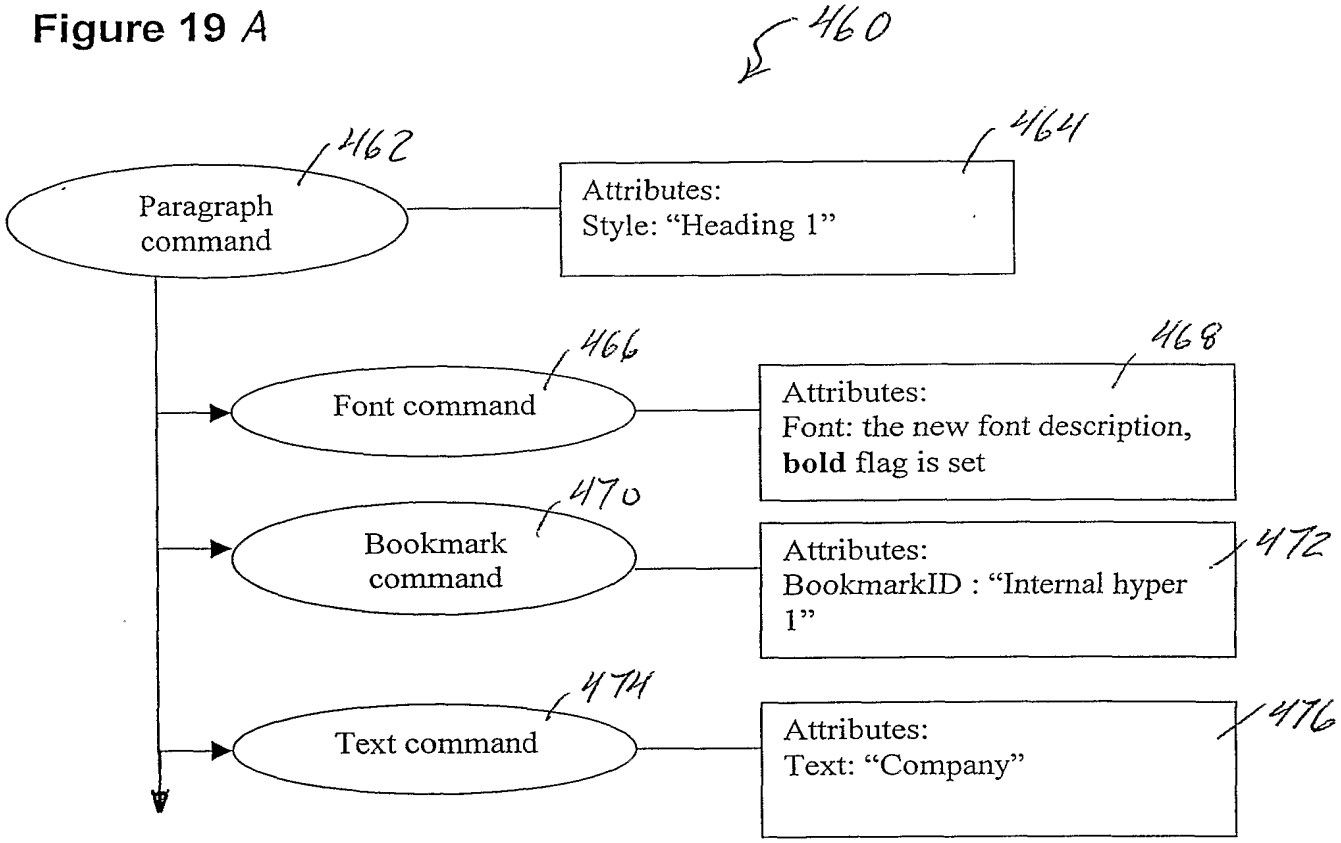

460
462
Paragraph command
464
Attributes:
Style: "Heading 1"
466
Font command
468
Attributes:
Font: the new font description,
bold flag is set
470
Bookmark command
472
Attributes:
BookmarkID : "Internal hyper 1"
474
Text command
476
Attributes:
Text: "Company"

Figure 19 B

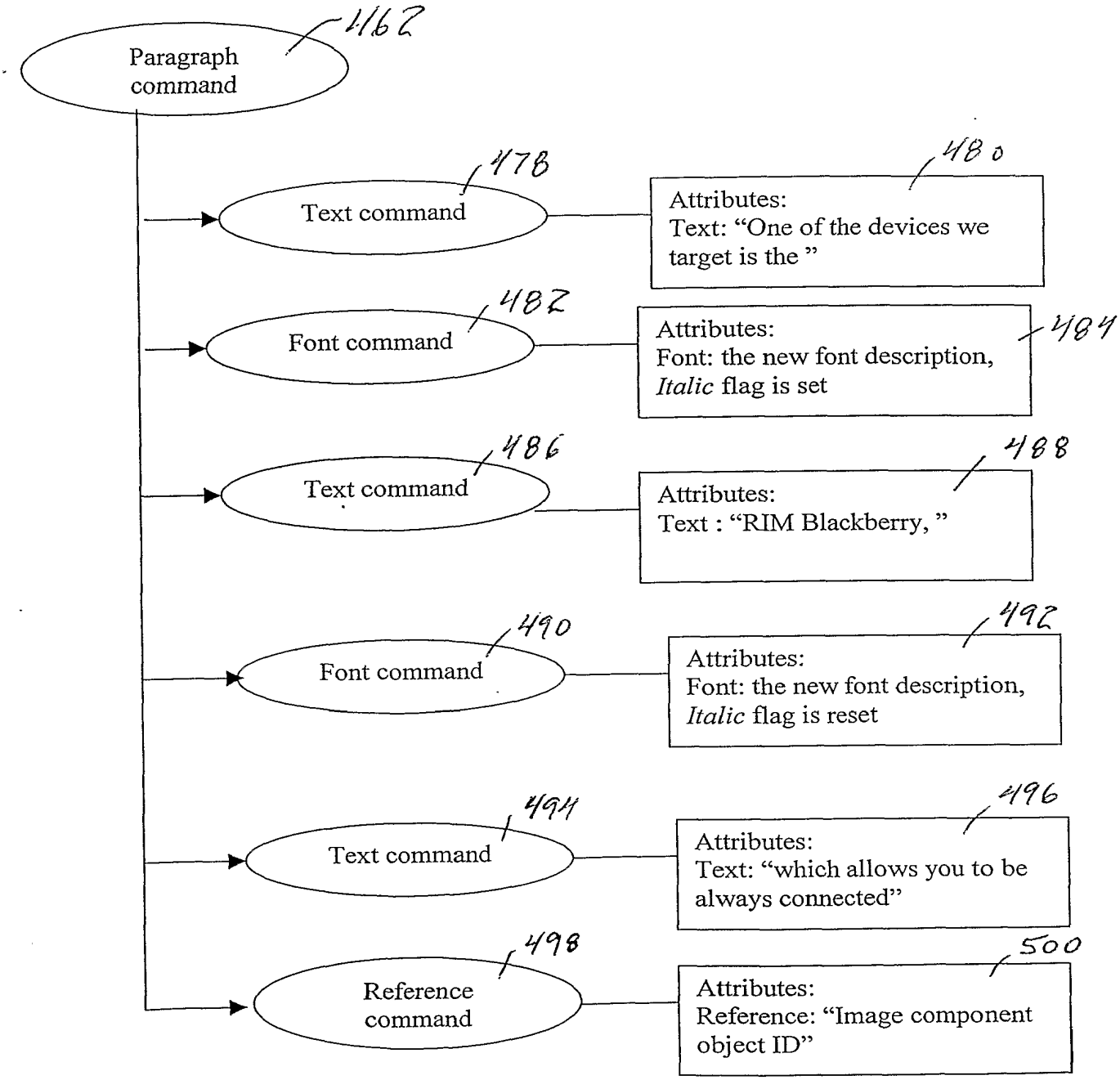

Paragraph command
462
Text command
478
Attributes:
Text: "One of the devices we target is the "
480
Font command
482
Attributes:
Font: the new font description, *Italic* flag is set
484
Text command
486
Attributes:
Text : "RIM Blackberry, "
488
Font command
490
Attributes:
Font: the new font description, *Italic* flag is reset
492
Text command
494
Attributes:
Text: "which allows you to be always connected"
496
Reference command
498
Attributes:
Reference: "Image component object ID"
500

Figure 20

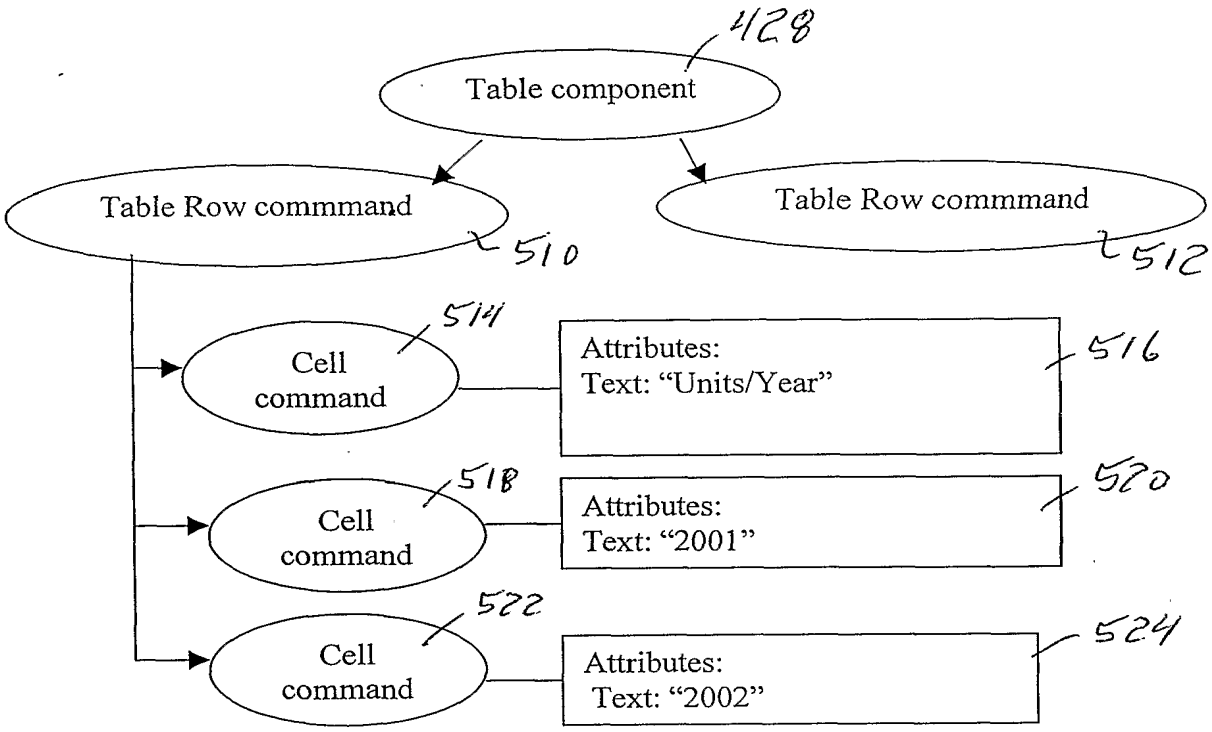

428
Table component
Table Row commmand
510
Table Row commmand
512
514
Cell command
Attributes:
Text: "Units/Year"
516
518
Cell command
Attributes:
Text: "2001"
520
522
Cell command
Attributes:
Text: "2002"
524

Figure 21A

380

Record ID	Structured Doc
1	UCS Signature
2	Document Header
3	Document Container
4	Paragraph
5	Paragraph Information
6	Bookmark
7	Simple Font Change
8	Text Segment
9	Paragraph
10	Paragraph Information
11	Paragraph
12	Paragraph Information
13	Hyperlink Begin
14	Simple Font Change
15	Text Segment
16	Simple Font Change
17	Text Segment
18	Text Segment
19	Hyperlink End
20	Paragraph
21	Paragraph Information
22	Hyperlink Begin
23	Simple Font Change
24	Text Segment
25	Simple Font Change
26	Text Segment
27	Text Segment
28	Hyperlink End
29	Paragraph
30	Paragraph Information
31	Bookmark
32	Hyperlink Begin
33	Simple Font Change
34	Text Segment
35	Simple Font Change
36	Text Segment
37	Text Segment
38	Hyperlink End
39	Paragraph
40	Paragraph Information

360

Record ID	Unstructured Doc
1	UCS Signature
2	Document Header
3	Section Container
4	Paragraph
5	Paragraph Information
6	Bookmark
7	Simple Font Change
8	Text Segment
9	Paragraph
10	Paragraph Information
11	Paragraph
12	Paragraph Information
13	Hyperlink Begin
14	Simple Font Change
15	Text Segment
16	Simple Font Change
17	Text Segment
18	Text Segment
19	Hyperlink End
20	Reference 1 - Text Box
21	Paragraph
22	Paragraph Information
23	Hyperlink Begin
24	Simple Font Change
25	Text Segment
26	Simple Font Change
27	Text Segment
28	Text Segment
29	Hyperlink End
30	Paragraph
31	Paragraph Information
32	Bookmark
33	Hyperlink Begin
34	Simple Font Change
35	Text Segment
36	Simple Font Change
37	Text Segment
38	Text Segment
39	Hyperlink End
40	Reference 2 - Image

Figure 21B

380

41	Hyperlink Begin
42	Simple Font Change
43	Text Segment
44	Simple Font Change
45	Text Segment
46	Text Segment
47	Hyperlink End
48	Paragraph
49	Paragraph Information
50	Hyperlink Begin
51	Simple Font Change
52	Text Segment
53	Simple Font Change
54	Text Segment
55	Text Segment
56	Hyperlink End
57	Paragraph
58	Paragraph Information
59	Paragraph
60	Paragraph Information
61	Simple Font Change
62	Text Segment
63	Paragraph
64	Paragraph Information
65	Simple Font Change
66	Text Segment
67	Hyperlink Begin
68	Simple Font Change
69	Text Segment
70	Hyperlink End
71	Simple Font Change
72	Text Segment
73	Hyperlink Begin
74	Simple Font Change
75	Text Segment
76	Bookmark
77	Text Segment
78	Hyperlink End
79	Simple Font Change
80	Text Segment
81	Paragraph
82	Paragraph Information
83	Reference 1 - Text Box
84	Reference 2 - Image
85	Simple Font Change

360

41	Paragraph
42	Paragraph Information
43	Hyperlink Begin
44	Simple Font Change
45	Text Segment
46	Simple Font Change
47	Text Segment
48	Text Segment
49	Hyperlink End
50	Paragraph
51	Paragraph Information
52	Hyperlink Begin
53	Simple Font Change
54	Text Segment
55	Simple Font Change
56	Text Segment
57	Text Segment
58	Hyperlink End
59	Paragraph
60	Paragraph Information
61	Paragraph
62	Paragraph Information
63	Simple Font Change
64	Text Segment
65	Paragraph
66	Paragraph Information
67	Simple Font Change
68	Text Segment
69	Hyperlink Begin
70	Simple Font Change
71	Text Segment
72	Hyperlink End
73	Simple Font Change
74	Text Segment
75	Hyperlink Begin
76	Simple Font Change
77	Text Segment
78	Bookmark
79	Text Segment
80	Hyperlink End
81	Simple Font Change
82	Text Segment
83	Paragraph
84	Paragraph Information
85	Simple Font Change

Figure 21C

380

86	Text Segment
87	Paragraph
88	Paragraph Information
89	Simple Font Change
90	Text Segment
91	Paragraph
92	Paragraph Information
93	Simple Font Change
94	Text Segment
95	Text Segment
96	Hyperlink Begin
97	Simple Font Change
98	Text Segment
99	Hyperlink End
100	Paragraph
101	Paragraph Information
102	Simple Font Change
103	Text Segment
104	Paragraph
105	Paragraph Information
106	Simple Font Change
107	Text Segment
108	Text Segment
109	Text Segment
110	Paragraph
111	Paragraph Information
112	Reference 3 - Image
113	Simple Font Change
114	Text Segment
115	Reference 4 - Spreadsheet
116	Paragraph
117	Paragraph Information
118	Simple Font Change
119	Text Segment
120	Paragraph
121	Paragraph Information
122	Paragraph
123	Paragraph Information
124	Simple Font Change
125	Text Segment
126	Text Segment
127	Paragraph
128	Paragraph Information
129	Simple Font Change
130	Text Segment

360

86	Text Segment
87	Paragraph
88	Paragraph Information
89	Simple Font Change
90	Text Segment
91	Paragraph
92	Paragraph Information
93	Simple Font Change
94	Text Segment
95	Text Segment
96	Hyperlink Begin
97	Simple Font Change
98	Text Segment
99	Hyperlink End
100	Paragraph
101	Paragraph Information
102	Simple Font Change
103	Text Segment
104	Paragraph
105	Paragraph Information
106	Simple Font Change
107	Text Segment
108	Text Segment
109	Text Segment
110	Paragraph
111	Paragraph Information
112	Simple Font Change
113	Text Segment
114	Paragraph
115	Paragraph Information
116	Simple Font Change
117	Text Segment
118	Paragraph
119	Paragraph Information
120	Paragraph
121	Paragraph Information
122	Simple Font Change
123	Text Segment
124	Text Segment
125	Paragraph
126	Paragraph Information
127	Simple Font Change
128	Text Segment
129	Paragraph
130	Paragraph Information

Figure 21D

380

131	Paragraph
132	Paragraph Information
133	Simple Font Change
134	Text Segment
135	Paragraph
136	Paragraph Information
137	Paragraph
138	Paragraph Information
139	Simple Font Change
140	Text Segment
141	Text Segment
142	Paragraph
143	Paragraph Information
144	Simple Font Change
145	Text Segment
146	Paragraph
147	Paragraph Information
148	Simple Font Change
149	Text Segment
150	Paragraph
151	Paragraph Information
152	Paragraph
153	Paragraph Information
154	Paragraph
155	Paragraph Information
156	Simple Font Change
157	Text Segment
158	Hyperlink Begin
159	Simple Font Change
160	Text Segment
161	Hyperlink End
162	Simple Font Change
163	Text Segment
164	Text Segment
165	Reference Containers
166	Reference 1
167	Reference 2
168	Reference 3
169	Reference 4
170	End of Document

360

131	Simple Font Change
132	Text Segment
133	Paragraph
134	Paragraph Information
135	Paragraph
136	Paragraph Information
137	Simple Font Change
138	Text Segment
139	Text Segment
140	Paragraph
141	Paragraph Information
142	Simple Font Change
143	Text Segment
144	Paragraph
145	Paragraph Information
146	Simple Font Change
147	Text Segment
148	Paragraph
149	Paragraph Information
150	Paragraph
151	Paragraph Information
152	Paragraph
153	Paragraph Information
154	Simple Font Change
155	Text Segment
156	Hyperlink Begin
157	Simple Font Change
158	Text Segment
159	UCS Signature
160	Document Header
161	Document Container
162	Paragraph
163	Paragraph Information
164	Bookmark
165	Simple Font Change
166	Text Segment
167	Paragraph
168	Paragraph Information
169	Simple Font Change
170	Text Segment
171	Paragraph
172	Paragraph Information
173	Paragraph
174	Paragraph Information
175	Bookmark

Figure 21E

360

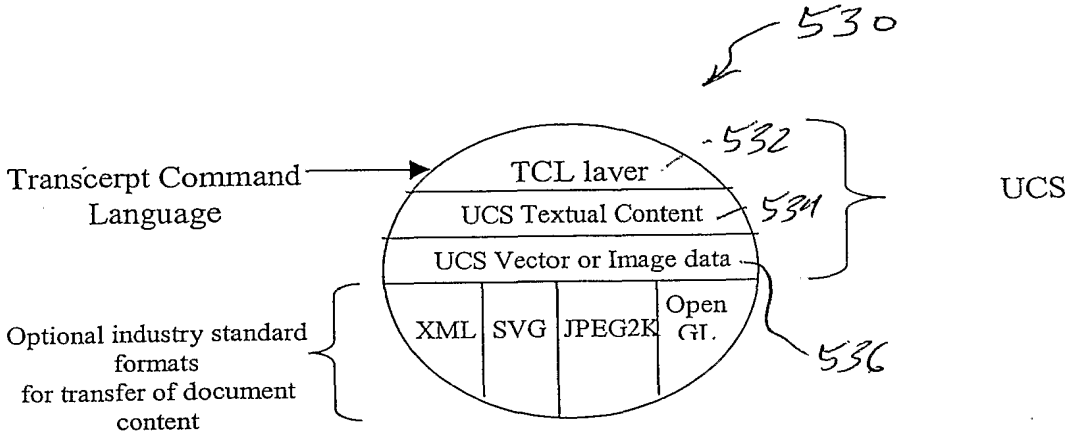
176	Simple Font Change
177	Text Segment
178	Paragraph
179	Paragraph Information
180	Paragraph
181	Paragraph Information
182	Bookmark
183	Bookmark
184	Simple Font Change
185	Text Segment
186	Paragraph
187	Paragraph Information
188	Simple Font Change
189	Text Segment
190	Paragraph
191	Paragraph Information
192	Bookmark
193	Simple Font Change
194	Text Segment
195	Paragraph
196	Paragraph Information
197	Paragraph
198	Paragraph Information
199	Simple Font Change
200	Text Segment
201	Paragraph
202	Paragraph Information
203	Simple Font Change
204	Text Segment
205	Simple Font Change
206	Text Segment
207	Simple Font Change
208	Text Segment
209	Simple Font Change
210	Text Segment
211	Paragraph
212	Paragraph Information
213	Simple Font Change
214	Text Segment
215	Simple Font Change
216	Text Segment
217	Simple Font Change
218	Text Segment
219	Simple Font Change
220	Text Segment

Figure 21F

360

221	Paragraph
222	Paragraph Information
223	Paragraph
224	Paragraph Information
225	Simple Font Change
226	Text Segment
227	Paragraph
228	Paragraph Information
229	Simple Font Change
230	Text Segment
231	Paragraph
232	Paragraph Information
233	Simple Font Change
234	Text Segment
235	Simple Font Change
236	Text Segment
237	Paragraph
238	Paragraph Information
239	Paragraph
240	Paragraph Information
241	Reference containers
242	Reference 1
243	Reference 2
244	End of Document

Figure 22


530
532
Transcerpt Command Language
TCL laver
UCS Textual Content
534
UCS Vector or Image data
UCS
Optional industry standard formats for transfer of document content
XML
SVG
JPEG2K
Open GL
536

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US02/09268

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06F 17/21
US CL : 707/513

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 707/513; 709/246

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
WEST USPAT

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A,E	US 6,404,445 B1 (GALEA et al) 11 June 2002 (11.06.2002).	1-15
A,E	US 6,401,132 B1 (BELLWOOD et al) 04 June 2002 (04.06.2002).	1-15
A,E	US 6,389,467 B1 (EYAL) 14 May 2002 (15.05.2002).	1-15
A,E	US 6,377,957 B1 (JEYARAMAN) 23 April 2002 (23.04.2002).	1-15
A,P	US 6,311,187 B1 (JEYARAMAN) 30 October 2001 (30.10.2001).	1-15

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
12 June 2002 (12.06.2002)

Date of mailing of the international search report
08 JUL 2002

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703)305-3230

Authorized officer
JOSEPH H FEILD
Peggy Harrod

Telephone No. (703)305-3900

Form PCT/ISA/210 (second sheet) (July 1998)